



Universidad
Carlos III de Madrid

Departamento de Informática

PROYECTO FIN DE CARRERA

PLATAFORMA DISTRIBUIDA PARA EL CONTROL DE PROYECTOS SOFTWARE

Autor: Adrián Hontoria Sánchez

Tutores: Luis Miguel Sánchez García

José Daniel García Sánchez

Colmenarejo, Junio de 2012



“Vale más saber alguna cosa de todo, que saberlo todo de una sola cosa”.
Blaise Pascal



Agradecimientos

En primer lugar agradecer a mi tutor Luis Miguel y al director José Daniel, por guiarme a lo largo del proyecto y corregir los errores durante este proceso, dedicándome parte de su tiempo.

A mis padres y a mi hermano por todo el apoyo que me han dado durante todos estos años de carrera.

A María por darme su opinión sincera y motivarme con el proyecto.

Resumen

La plataforma creada, llamada CPS (Control Distribuido de Proyectos Software), es una plataforma distribuida que permite disponer de la información directamente relacionada con el desarrollo de un proyecto desde cualquier dispositivo con acceso a internet. El programa CPS permite llevar a cabo una amplia serie de acciones cuyo objetivo principal es controlar el desarrollo de un proyecto de una forma simplificada y unificada.

CPS forma parte de un proyecto global, el cual se divide en dos ramas: gestión de usuarios y gestión de proyectos. Ésta última es abarcada completamente por CPS.

Los usuarios que pertenezcan a la primera rama, tendrán acceso a CPS dependiendo de su perfil. En dicho perfil encontrarán su rol general, los proyectos en los que participa y el rol específico para cada proyecto. CPS hace uso de dicha información para clasificar así diferentes opciones y áreas dependiendo del usuario que accede al sistema.

Los usuarios, dependiendo del rol que tengan preestablecido, podrán crear proyectos, editarlos, gestionar requisitos, casos de uso, gestionar la trazabilidad, crear entregables, etc., todo ello, dentro del mismo entorno.

Gracias a la lectura del perfil de usuario que hace CPS, un equipo de proyecto podrá desarrollar el proyecto teniendo un contacto directo con el cliente, ya que la plataforma dispone de un módulo de publicaciones, de tal forma que el cliente participe de una forma directa en el desarrollo aceptando o rechazando diferentes elementos publicados. Cada elemento individual de los proyectos dispone de un ciclo de vida, desde su creación hasta su aceptación, cancelación o borrado.

Todo desarrollo de proyecto debe seguir una metodología para asegurar un mínimo de calidad y consistencia. CPS orienta los desarrollos mediante la metodología RUP (Rational Unified Process). Dicha metodología proporciona un enfoque disciplinado para la asignación de tareas y responsabilidades dentro de una organización de desarrollo, con el fin de garantizar la producción de software de alta calidad que satisfaga las necesidades de sus usuarios finales. CPS hace uso de dicha asignación para así tener un control absoluto de los elementos que van dando forma al proyecto a desarrollar.

Pero no significa que el desarrollador quede obligado a utilizar la metodología RUP. CPS dispone de un amplio abanico de posibilidades totalmente abiertas para no forzar el desarrollo.

Palabras clave: control, distribuido, proyectos, software, CPS, RUP, requisitos, casos de uso, entregables, roles, equipo, trazabilidad, gestión, desarrollo.

Índice

1. INTRODUCCIÓN.....	16
1.1. MOTIVACIÓN	16
1.2. OBJETIVO	17
1.3. ESQUEMA DEL DOCUMENTO	18
2. ESTADO DE LA CUESTIÓN	19
2.1. SISTEMAS DE CONTROL DE PROYECTOS	19
2.1.1 IBM Rational Doors	19
2.1.2 IBM Rational RequisitePro	21
2.1.3 PHProjekt	22
2.1.4 Microsoft Project	23
2.2. TECNOLOGÍAS WEB	25
2.2.1 Tecnologías de desarrollo de aplicaciones web	25
2.2.2 Sistemas gestores de bases de datos	31
3. ANÁLISIS Y DISEÑO.....	35
3.1. PLANIFICACIÓN DEL PROYECTO	35
3.2. PLAN DE ITERACIONES	38
3.3. CASOS DE USO	40
CU1: Gestionar proyectos.....	41
CU2: Gestionar plan de iteraciones.....	42
CU3: Gestionar requisitos.....	46
CU4: Gestionar casos de uso	48
CU5: Gestionar trazabilidad	51
CU6: Gestionar entregables.....	52
CU7: Personalizar gráficas.....	54
CU8: Gestionar avisos	55
3.4. REQUISITOS SUPLEMENTARIOS	56
3.5. MATRIZ DE TRAZABILIDAD.....	79
3.6. ANÁLISIS DE LA BASE DE DATOS	82
3.6.1 Entidad Proyecto.....	84
3.6.2 Entidad Entregable.....	85
3.6.3 Entidad Caso de uso	85
3.6.4 Entidad Requisito	85
3.6.5 Entidad Iteración.....	86
3.6.6 Entidades Histórico	86
3.6.7 Entidad Usuario	86
3.6.8 Entidad Rol	87
3.6.9 Entidad Gráfica	87
3.7. DIAGRAMA DE FLUJO POR BLOQUES	88
3.7.1 Módulo Login	89
3.7.2 Módulo Menú principal	89
3.7.3 Módulo Proyectos	90
3.7.4 Módulo Personalización	91
3.7.5 Módulo Ayuda.....	91
3.7.6 Módulo Avisos.....	91
3.7.7 Módulo Iteraciones	92
3.7.8 Módulo Requisitos/Casos de uso	92
3.7.9 Módulo Trazabilidad	93

3.7.10	Módulo Entregables.....	94
3.7.11	Módulo Gestión Personalizar	94
3.7.12	Módulo Gestión Proyectos	95
3.7.13	Módulo Gestión Avisos.....	95
3.7.14	Módulo Gestión Iteraciones	96
3.7.15	Módulo Gestión Requisitos	96
3.7.16	Módulo Gestión Trazabilidad	97
3.7.17	Módulo Gestión Entregables.....	97
3.7.18	Módulo Gestión Publicar.....	98
3.7.19	Módulo Gestión Eliminar.....	98
3.8.	DISPOSICIÓN DE LAS PÁGINAS	100
3.8.1	Diseño de tipo "Login"	100
3.8.2	Diseño de tipo "Menú principal"	101
3.8.3	Diseño de tipo "Listados".....	102
3.8.4	Diseño de tipo "Formularios".....	103
3.8.5	Diseño de tipo "Principal de un proyecto"	104
3.9.	ESTRUCTURA DE LA BASE DE DATOS	105
3.9.1	Entidad Proyectos	106
3.9.2	Entidad Usuarios.....	106
3.9.3	Entidad Roles	107
3.9.4	Entidad Usuarios_participantes.....	107
3.9.5	Entidad Historico_participantes.....	107
3.9.6	Entidad Graficas.....	108
3.9.7	Entidad Personalizacion	108
3.9.8	Entidad Avisos.....	109
3.9.9	Entidad Iteraciones	109
3.9.10	Entidad Fases	110
3.9.11	Entidad Productos	110
3.9.12	Entidad Iteracion_producto	110
3.9.13	Entidad Historico_iteraciones.....	111
3.9.14	Entidad Entregables.....	111
3.9.15	Entidad Requisitos	112
3.9.16	Entidad Casos_uso.....	113
3.9.17	Entidad Trazabilidad	113
3.9.18	Entidad Historico_requisitos	113
3.9.19	Entidad Historico_casos_uso.....	114
3.9.20	Entidad Requisitos_proyecto.....	115
3.9.21	Entidad Subrequisitos_proyecto.....	115
3.9.22	Entidad Tipos_requisitos.....	116
3.9.23	Entidad Subtipos_requisitos.....	116
3.9.24	Entidad Niveles.....	116
4.	DESARROLLO.....	117
4.1.	MÓDULO LOGIN	117
4.1.1	install.php	117
4.1.2	index.php	119
4.2.	MÓDULO MENÚ PRINCIPAL	120
4.2.1	index.php	120
4.3.	MÓDULO PROYECTOS	121
4.3.1	proyectos.php	121
4.3.2	index_proyecto.php.....	121
4.3.3	crear_proyecto.php.....	122
4.3.4	editar_proyecto.php.....	123
4.3.5	grafica_bar1.php, grafica_bar2.php y grafica_pie.php.....	123
4.3.6	asignarProyecto.php	125
4.3.7	gestion_zip.php.....	125
4.3.8	historial_proyectos.php.....	127

4.4.	MÓDULO PERSONALIZACIÓN	128
4.4.1	<i>personalizar.php</i>	128
4.5.	MÓDULO AYUDA	129
4.6.	MÓDULO AVISOS	130
4.6.1	<i>avisos.php</i>	130
4.6.2	<i>crear_aviso.php</i>	131
4.6.3	<i>leer_aviso.php</i>	132
4.6.4	<i>responder_aviso.php</i>	132
4.7.	MÓDULO ITERACIONES	133
4.7.1	<i>iteraciones.php</i>	133
4.7.2	<i>crear_iteracion.php</i>	134
4.7.3	<i>editar_iteracion.php</i>	134
4.7.4	<i>historial_iteraciones.php</i>	136
4.8.	MÓDULO REQUISITOS/CASOS DE USO	137
4.8.1	<i>requisitos.php</i>	137
4.8.2	<i>crear_requisito.php</i>	138
4.8.3	<i>editar_requisito.php</i>	139
4.8.4	<i>editar_caso.php</i>	141
4.8.5	<i>historial_requisitos.php</i>	142
4.9.	MÓDULO TRAZABILIDAD	143
4.9.1	<i>trazabilidad.php</i>	143
4.9.2	<i>editar_trazabilidad.php</i>	144
4.10.	MÓDULO ENTREGABLES	145
4.10.1	<i>entregables.php</i>	145
4.10.2	<i>crear_entregable.php</i>	146
4.10.3	<i>editar_entregable.php</i>	147
4.11.	MÓDULO GESTIÓN PERSONALIZAR	148
4.11.1	<i>gestion_personalizar.php</i>	148
4.12.	MÓDULO GESTIÓN PROYECTOS	149
4.12.1	<i>gestion_proyectos.php</i>	149
4.13.	MÓDULO GESTIÓN AVISOS	150
4.13.1	<i>gestion_avisos.php</i>	150
4.14.	MÓDULO GESTIÓN ITERACIONES	151
4.14.1	<i>gestion_iteraciones.php</i>	151
4.15.	MÓDULO GESTIÓN REQUISITOS	152
4.15.1	<i>gestion_requisitos.php</i>	152
4.16.	MÓDULO GESTIÓN TRAZABILIDAD	153
4.16.1	<i>gestion_trazabilidad.php</i>	153
4.17.	MÓDULO GESTIÓN ENTREGABLES	154
4.17.1	<i>gestion_entregables.php</i>	154
4.18.	MÓDULO GESTIÓN PUBLICAR	155
4.18.1	<i>gestion_publicar.php</i>	155
4.18.2	<i>gestion_publicar_entregable.php</i>	157
4.19.	MÓDULO GESTIÓN ELIMINAR	159
4.19.1	<i>gestion_eliminar.php</i>	159
4.20.	FUNCIONES AUXILIARES	159
5.	CONCLUSIONES	160
5.1.	CONCLUSIONES GENERALES	160
5.2.	TRABAJOS FUTUROS	161
5.3.	PRESUPUESTO	162
6.	GLOSARIO	163
7.	REFERENCIAS	170



ANEXO I: PRESUPUESTO.....	171
ANEXO II: MANUAL DE INSTALACIÓN.....	173
ANEXO III: MANUAL DE USUARIO.	175
ANEXO IV: FUNCIONES AUXILIARES.....	203
ANEXO V: EJEMPLO PRÁCTICO DE USO.	215

Índice de figuras

FIGURA 1. GESTIÓN DE REQUISITOS EN IBM RATIONAL DOORS.	20
FIGURA 2. MATRIZ DE TRAZABILIDAD EN IBM RATIONAL REQUISITEPRO.....	21
FIGURA 3. GESTIÓN DE PLANIFICACIÓN EN PHPROJEKT.	22
FIGURA 4. GESTIÓN DE PLANIFICACIÓN EN MICROSOFT PROJECT.	23
FIGURA 5. MODELO LAMP.....	34
FIGURA 6. DIAGRAMA DE GANTT.	37
FIGURA 7. FUNCIONALIDAD GENERAL DEL SISTEMA.	40
FIGURA 8. CU1: GESTIONAR PROYECTOS.....	41
FIGURA 9. CU2: GESTIONAR PLAN DE ITERACIONES.	43
FIGURA 10. CU3: GESTIONAR REQUISITOS.	46
FIGURA 11. CU4: GESTIONAR CASOS DE USO.	49
FIGURA 12. CU5: GESTIONAR TRAZABILIDAD.	51
FIGURA 13. CU6: GESTIONAR ENTREGABLES.	52
FIGURA 14. CU7: PERSONALIZAR GRÁFICAS.	54
FIGURA 15. CU8: GESTIONAR AVISOS.....	55
FIGURA 16. DIAGRAMA ENTIDAD RELACIÓN.....	83
FIGURA 17. DIAGRAMA DE FLUJO DEL SISTEMA.....	88
FIGURA 18. DISEÑO DE TIPO “LOGIN”.	100
FIGURA 19. DISEÑO DE TIPO “MENÚ PRINCIPAL”	101
FIGURA 20. DISEÑO DE TIPO “LISTADOS”.	102
FIGURA 21. DISEÑO DE TIPO “FORMULARIOS”.....	103
FIGURA 22. DISEÑO DE TIPO “PRINCIPAL DE UN PROYECTO”.	104
FIGURA 23. ESTRUCTURA DE LA BASE DE DATOS.....	105

Índice de tablas

TABLA 1. COMPARATIVA DE TECNOLOGÍAS WEB.	30
TABLA 2. COMPARATIVA DE SISTEMAS GESTORES DE BASES DE DATOS.	33
TABLA 3. PLAN DE ITERACIONES.....	38
TABLA 4. ITERACIÓN 1: FUNCIONES BÁSICAS.	38
TABLA 5. ITERACIÓN 2: FUNCIONES ADICIONALES.	39
TABLA 6. ITERACIÓN 3: GRÁFICAS Y DOCUMENTOS.	39
TABLA 7. ITERACIÓN 4: FUNCIONES EXTRA.	39
TABLA 8. R1: GESTIONAR PROYECTOS.....	57
TABLA 9. R2: GESTIONAR ITERACIONES.....	57
TABLA 10. R3: GESTIONAR REQUISITOS.....	57
TABLA 11. R4: GESTIONAR CASOS DE USO.....	57
TABLA 12. R5: GESTIONAR TRAZABILIDAD.....	58
TABLA 13. R6: GESTIONAR ENTREGABLES.	58
TABLA 14. R7: DATOS CASOS DE USO.	58
TABLA 15. R8: GENERACIÓN DE ENTREGABLES.....	58
TABLA 16. R9: GENERACIÓN MANUAL DE ENTREGABLES.	59
TABLA 17. R10: GENERACIÓN AUTOMÁTICA DE ENTREGABLES.	59
TABLA 18. R11: FORMATO DE DOCUMENTOS.	59
TABLA 19. R12: FORMATO DE ENTREGABLES.	59
TABLA 20. R13: DATOS DE ITERACIONES.	60
TABLA 21. R14: DATOS DE PROYECTOS.	60
TABLA 22. R15: DATOS DE REQUISITOS.	60
TABLA 23. R16: TIPOS DE REQUISITOS.....	60
TABLA 24. R17: SUBTIPOS DE REQUISITOS.	61
TABLA 25. R18: INTERFAZ DE USUARIOS.	61
TABLA 26. R19: CLASIFICACIÓN DE USUARIOS.....	61
TABLA 27. R20: GESTIONAR PARTICIPANTES EN UN PROYECTO.	61
TABLA 28. R21: LISTADO DE PROYECTOS.....	62
TABLA 29. R22: LISTADO DE ITERACIONES.....	62
TABLA 30. R23: LISTADO DE REQUISITOS.....	62
TABLA 31. R24: LISTADO DE CASOS DE USO.	62
TABLA 32. R25: LISTADO DE ENTREGABLES.	63
TABLA 33. R26: LISTADO DE AVISOS.....	63



TABLA 34. R27: PUBLICAR REQUISITO.	63
TABLA 35. R28: PUBLICAR CASO DE USO.	63
TABLA 36. R29: PUBLICAR ENTREGABLE.	64
TABLA 37. R30: ESTADOS DE ELEMENTOS.	64
TABLA 38. R31: ESTADO PROCESADO.	64
TABLA 39. R32: ESTADO REVISANDO.	64
TABLA 40. R33: ESTADO PUBLICADO.	65
TABLA 41. R34: ESTADO RECHAZADO.	65
TABLA 42. R35: ESTADO ACEPTADO.	65
TABLA 43. R36: ACEPTAR REQUISITO.	66
TABLA 44. R37: ACEPTAR CASO DE USO.	66
TABLA 45. R38: ACEPTAR ENTREGABLE.	66
TABLA 46. R39: RECHAZAR REQUISITO.	66
TABLA 47. R40: RECHAZAR CASO DE USO.	67
TABLA 48. R41: RECHAZAR ENTREGABLE.	67
TABLA 49. R42: GESTIONAR AVISOS.	67
TABLA 50. R43: DATOS DE AVISOS.	67
TABLA 51. R44: DESCARGAR PROYECTO.	68
TABLA 52. R45: HISTORIAL DEL PROYECTO.	68
TABLA 53. R46: GRÁFICAS DE UN PROYECTO.	68
TABLA 54. R47: GRÁFICA "NÚMERO DE REQUISITOS SUPLEMENTARIOS POR ESTADO".	68
TABLA 55. R48: GRÁFICA "NÚMERO DE ITERACIONES POR ESTADO Y FASES".	69
TABLA 56. R49: GRÁFICA "REQUISITOS EDITADOS POR SEMANA".	69
TABLA 57. R50: GRÁFICA "ESTABILIDAD MEDIA DE REQUISITOS POR SEMANA".	69
TABLA 58. R51: GRÁFICA "NÚMERO DE ITERACIONES POR ESTADO".	69
TABLA 59. R52: GRÁFICA "CASOS DE USO POR ESTADO".	70
TABLA 60. R53: GRÁFICA "ENTREGABLES POR ESTADO".	70
TABLA 61. R54: GRÁFICA "CASOS DE USO EDITADOS POR SEMANA".	70
TABLA 62. R55: GRÁFICA "ENTREGABLES EDITADOS POR SEMANA".	70
TABLA 63. R56: GRÁFICA "REQUISITOS POR VERSIÓN".	71
TABLA 64. R57: GRÁFICA "CASOS DE USO POR VERSIÓN".	71
TABLA 65. R58: GRÁFICA "ENTREGABLES POR VERSIÓN".	71
TABLA 66. R59: GESTIÓN DE GRÁFICAS.	71
TABLA 67. R60: AYUDA AL USUARIO.	72
TABLA 68. R61: HISTORIAL DEL REQUISITO.	72
TABLA 69. R62: HISTORIAL DEL CASO DE USO.	72
TABLA 70. R63: PERMISO DE ADMINISTRADOR.	72
TABLA 71. R64: PERMISO DE PROPIETARIO.	73

TABLA 72. R65: PERMISO DE PROJECT MANAGER.	73
TABLA 73. R66: PERMISO DE INGENIERO DE REQUISITOS.....	73
TABLA 74. R67: PERMISO DE GESTOR DE ENTREGABLES.....	73
TABLA 75. R68: SEGURIDAD FRENTE ATAQUES.....	74
TABLA 76. R69: CONTROL DE ACCESO.	74
TABLA 77. R70: COMPATIBILIDAD CON NAVEGADORES WEB.....	74
TABLA 78. R71: GARANTIZAR EL CUMPLIMIENTO DE LA LEY ORGÁNICA DE PROTECCIÓN DE DATOS.	74
TABLA 79. R72: DISEÑO DE LA APLICACIÓN.	75
TABLA 80. R73: MENÚ PRINCIPAL.....	75
TABLA 81. R74: ESTRUCTURA DE LAS SECCIONES.	75
TABLA 82. R75: SELECTOR DE PROYECTOS.....	75
TABLA 83. R76: SELECTOR DE AVISOS.....	76
TABLA 84. R77: ESTRUCTURA DE TABLAS.	76
TABLA 85. R78: ESTRUCTURA DE LA PANTALLA PRINCIPAL DE UN PROYECTO.....	76
TABLA 86. R79: OPTIMIZACIÓN PARA NAVEGADOR CHROME.	76
TABLA 87. R80: LOGOTIPO DE LA APLICACIÓN.....	77
TABLA 88. R81: HARDWARE DE USUARIO.....	77
TABLA 89. R82: RESTRICCIONES DE USUARIO.	77
TABLA 90. R83: RESTRICCIONES DEL SERVIDOR.	78
TABLA 91. MATRIZ DE TRAZABILIDAD.	81
TABLA 92. ENTIDAD PROYECTOS.	106
TABLA 93. ENTIDAD USUARIOS.	106
TABLA 94. ENTIDAD ROLES.....	107
TABLA 95. ENTIDAD USUARIOS_PARTICIPANTES.....	107
TABLA 96. ENTIDAD HISTORICO_PARTICIPANTES.....	108
TABLA 97. ENTIDAD GRAFICAS.	108
TABLA 98. ENTIDAD PERSONALIZACION.....	108
TABLA 99. ENTIDAD AVISOS.	109
TABLA 100. ENTIDAD ITERACIONES.....	109
TABLA 101. ENTIDAD FASES.	110
TABLA 102. ENTIDAD PRODUCTOS.	110
TABLA 103. ENTIDAD ITERACION_PRODUCTO.....	110
TABLA 104. ENTIDAD HISTORICO_ITERACIONES.....	111
TABLA 105. ENTIDAD ENTREGABLES.....	112
TABLA 106. ENTIDAD REQUISITOS.....	112
TABLA 107. ENTIDAD CASOS_USO.....	113
TABLA 108. ENTIDAD TRAZABILIDAD.....	113
TABLA 109. ENTIDAD HISTORICO_REQUISITOS.....	114



TABLA 110. ENTIDAD HISTORICO_CASOS_USO.....	115
TABLA 111. ENTIDAD REQUISITOS_PROYECTO.....	115
TABLA 112. ENTIDAD SUBREQUISITOS_PROYECTO.....	115
TABLA 113. ENTIDAD TIPOS_REQUISITOS.....	116
TABLA 114. ENTIDAD SUBTIPOS_REQUISITOS.	116
TABLA 115. ENTIDAD NIVELES.	116
TABLA 116. FUNCIONES AUXILIARES.	214

Índice de ilustraciones

ILUSTRACIÓN 1. INSTALADOR DE LA APLICACIÓN.	118
ILUSTRACIÓN 2. ACCESO A LA APLICACIÓN.	119
ILUSTRACIÓN 3. MENÚ PRINCIPAL.	120
ILUSTRACIÓN 4. LISTADO DE PROYECTOS.....	121
ILUSTRACIÓN 5. PANTALLA PRINCIPAL DEL PROYECTO.	122
ILUSTRACIÓN 6. CREACIÓN DE UN NUEVO PROYECTO.	123
ILUSTRACIÓN 7. EDICIÓN DE UN PROYECTO EXISTENTE.	123
ILUSTRACIÓN 8. GRÁFICA DE BARRAS DE TIPO 1.	124
ILUSTRACIÓN 9. GRÁFICA DE BARRAS DE TIPO 2.	124
ILUSTRACIÓN 10. GRÁFICA CIRCULAR.....	124
ILUSTRACIÓN 11. ASIGNACIÓN DE USUARIOS.	125
ILUSTRACIÓN 12. DESCARGA DE UN PROYECTO.	125
ILUSTRACIÓN 13. DESCARGA REMOTA DE UN PROYECTO.	126
ILUSTRACIÓN 14. HISTORIAL DEL PROYECTO.....	127
ILUSTRACIÓN 15. ORDENACIÓN DE DATOS.	127
ILUSTRACIÓN 16. ACTIVACIÓN Y DESACTIVACIÓN DE GRÁFICAS.	128
ILUSTRACIÓN 17. AYUDA A LOS USUARIOS.	129
ILUSTRACIÓN 18. AVISOS.	130
ILUSTRACIÓN 19. CREAR AVISO.....	131
ILUSTRACIÓN 20. LEER AVISO.....	132
ILUSTRACIÓN 21. RESPONDER AVISO.	132
ILUSTRACIÓN 22. LISTADO DE ITERACIONES.....	133
ILUSTRACIÓN 23. CREAR ITERACIÓN.	134
ILUSTRACIÓN 24. EDITAR ITERACIÓN.	135
ILUSTRACIÓN 25. HISTORIAL DE LA ITERACIÓN.	136
ILUSTRACIÓN 26. LISTADO DE REQUISITOS.	137
ILUSTRACIÓN 27. LISTADO DE CASOS DE USO.	138
ILUSTRACIÓN 28. CREAR CASO DE USO.	139
ILUSTRACIÓN 29. CREAR REQUISITO.	139
ILUSTRACIÓN 30. EDITAR REQUISITO.	140
ILUSTRACIÓN 31. EDITAR CASO DE USO.	141
ILUSTRACIÓN 32. HISTORIAL DE REQUISITOS.	142
ILUSTRACIÓN 33. MATRIZ DE TRAZABILIDAD.	143
ILUSTRACIÓN 34. EDICIÓN DE LA MATRIZ DE TRAZABILIDAD.	144



ILUSTRACIÓN 35. LISTADO DE ENTREGABLES.	145
ILUSTRACIÓN 36. CREAR OTRO ENTREGABLE.	146
ILUSTRACIÓN 37. ACTUALIZAR ENTREGABLE.	147
ILUSTRACIÓN 38. ACCESO AL SISTEMA.	176
ILUSTRACIÓN 39. MENÚ PRINCIPAL.	176
ILUSTRACIÓN 40. CREAR UN PROYECTO.	176
ILUSTRACIÓN 41. PANTALLA PRINCIPAL DE UN PROYECTO.	177
ILUSTRACIÓN 42. ASIGNACIÓN DE USUARIOS.	178
ILUSTRACIÓN 43. CREAR UNA ITERACIÓN.	181
ILUSTRACIÓN 44. VER ITERACIONES.	181
ILUSTRACIÓN 45. CREAR UN REQUISITO SUPLEMENTARIO.	183
ILUSTRACIÓN 46. CREAR UN CASO DE USO.	185
ILUSTRACIÓN 47. MATRIZ DE TRAZABILIDAD.	185
ILUSTRACIÓN 48. CREAR UN ENTREGABLE.	192
ILUSTRACIÓN 49. EJEMPLO DE GRÁFICA DE REQUISITOS SUPLEMENTARIOS POR ESTADO.	195
ILUSTRACIÓN 50. EJEMPLO DE GRÁFICA ITERACIONES POR ESTADO Y FASES.	196
ILUSTRACIÓN 51. EJEMPLO DE GRÁFICA DE EDICIONES DE REQUISITOS POR SEMANA.	196
ILUSTRACIÓN 52. EJEMPLO DE GRÁFICA DE ESTABILIDAD DE REQUISITOS POR SEMANA.	196
ILUSTRACIÓN 53. EJEMPLO DE GRÁFICA DE ITERACIONES POR ESTADO.	197
ILUSTRACIÓN 54. EJEMPLO DE GRÁFICA DE CASOS DE USO POR ESTADO.	197
ILUSTRACIÓN 55. EJEMPLO DE GRÁFICA DE ENTREGABLES POR ESTADO.	197
ILUSTRACIÓN 56. EJEMPLO DE GRÁFICA DE EDICIONES DE CASOS DE USO POR SEMANA.	197
ILUSTRACIÓN 57. EJEMPLO DE GRÁFICA DE EDICIONES DE ENTREGABLES POR SEMANA.	198
ILUSTRACIÓN 58. EJEMPLO DE GRÁFICA DE REQUISITOS POR VERSIÓN.	198
ILUSTRACIÓN 59. EJEMPLO DE GRÁFICA DE CASOS DE USO POR VERSIÓN.	198
ILUSTRACIÓN 60. EJEMPLO DE GRÁFICA DE ENTREGABLES POR VERSIÓN.	198
ILUSTRACIÓN 61. AVISOS RECIBIDOS.	201
ILUSTRACIÓN 62. DESARROLLO DE UN PROYECTO REAL.	215
ILUSTRACIÓN 63. CONTROL DEL DESARROLLO DE UN PROYECTO REAL.	215
ILUSTRACIÓN 64. CONTROL DEL TRABAJO REALIZADO.	216
ILUSTRACIÓN 65. PRODUCTOS GENERADOS DEL PROYECTO.	216

1. Introducción

A continuación se expone una breve introducción, comenzando por aquellos factores que han motivado la elección de este proyecto, continuando con los objetivos que pretende conseguir y terminando por una visión general del presente documento.

1.1. Motivación

El desarrollo de proyectos software es una tarea cada vez más compleja, debido a la cantidad de información que se procesa para llegar a unos resultados tangibles. Un ejemplo de la cantidad de información que se maneja durante el desarrollo de un sistema software se puede encontrar fácilmente en la documentación que se genera siguiendo algunas de las metodologías de desarrollo más complejas, como [RUP](#) o [Métrica v3](#).

Dichas metodologías dividen el proceso de creación de software en varias actividades, fases o tareas dependiendo de la metodología. Por cada actividad o fase es necesario generar un conjunto de documentos que permitan corroborar el objetivo que se pretendía conseguir en un punto concreto del desarrollo.

Teniendo en cuenta el aumento de la cantidad de información y el amplio número de participantes en un proyecto, sumados a una carencia de control durante todo el ciclo de vida en el desarrollo del proyecto, hace que se utilicen multitud de plataformas diferentes dando lugar a un desarrollo complejo del proyecto.

Aunque ya existe un gran número de sistemas de control de proyectos, ninguno de ellos unifica totalmente de una forma flexible y sencilla la parte técnica (requisitos, casos de uso...) junto con la parte comunicativa con el cliente, cuyo coste sea asumible por una pequeña o mediana empresa.

Hoy en día, la flexibilidad mencionada se relaciona directamente con la complejidad del acceso a la información. Es por ello por lo que el concepto “[cloud computing](#)” está cada vez más presente. Los sistemas tradicionales obligaban al usuario a consumir recursos de su máquina local mientras que gracias a las aplicaciones en la nube, los usuarios no tienen por qué preocuparse de sus ordenadores locales. Simplemente acceden a la información cuando lo necesiten y desde cualquier dispositivo con acceso a internet.

Si se tienen en cuenta todos los aspectos anteriores sin contar con el concepto “cloud computing”, nos encontramos con la siguiente situación: gran cantidad de información gestionada por un gran número de usuarios a través de aplicaciones que dependen de cada usuario. Esto da lugar a que el control sobre las tareas realizadas no sea el deseable.

El sistema a desarrollar permitirá un control absoluto de los proyectos gracias a que toda la información se encontrará en la nube, accesible desde cualquier dispositivo con conexión a internet.

A continuación se describen en profundidad los objetivos que pretende cumplir la nueva plataforma y se mostrará una breve estructura del presente documento.

1.2. Objetivo

El objetivo principal de este proyecto es el análisis, diseño e implementación de una plataforma distribuida que permita a los equipos de desarrollo de proyectos software unificar toda la información que surja durante todas las fases del desarrollo para mantener un control absoluto del proyecto, de principio a fin, siguiendo la filosofía de “[cloud computing](#)”.

Dicha información abarca un amplio abanico de posibilidades, que se traducen en los subobjetivos de la aplicación a desarrollar. Por lo tanto, el sistema permitirá:

- La gestión de proyectos de forma completa, orientando los desarrollos a la metodología [RUP](#).
- Controlar todas las actividades o tareas que se llevan a cabo sobre un proyecto.
- Clasificar a los participantes de un proyecto dependiendo del rol que ejerzan sobre el mismo.
- Gestionar el Plan de Iteraciones de un proyecto.
- Definir los casos de uso y requisitos de un proyecto, así como la trazabilidad entre ambos.
- Gestionar los productos o entregables de un proyecto.
- Publicar los elementos de un proyecto a los usuarios con el rol de propietario del proyecto en desarrollo.
- Mantener un contacto directo entre todos los participantes del desarrollo.
- Personalizar la visualización del estado de cada proyecto a cada usuario.

Hay que destacar que el objetivo relacionado con la gestión de usuarios no está contemplado, debido a que la aplicación a desarrollar pertenece a un proyecto más amplio, en el que las dos ramas principales son la gestión de proyectos y la gestión de usuarios.

En este mismo documento se podrá seguir la evolución de la aplicación, en lo que a ingeniería del software se refiere.

1.3. Esquema del documento

El documento comienza por un repaso al [contexto actual](#) de las plataformas de control de proyectos, explicando las características principales y las carencias de aquellas con más similitud respecto al sistema en desarrollo:

- IBM Rational Doors.
- IBM Rational RequisitePro.
- PHProjekt.
- Microsoft Project.

De igual forma, se realiza un análisis a las [tecnologías actuales](#) que permitirían ejecutar un desarrollo óptimo del sistema. Para ello se realiza un breve análisis sobre las tecnologías web y los sistemas gestores de bases de datos y se realiza una comparación entre los mismos.

A continuación, comienza la [planificación del proyecto](#) junto con la especificación y análisis de [casos de uso](#) y [requisitos suplementarios](#). Tras analizar la funcionalidad del sistema gracias a los requisitos y casos de uso, se realiza un análisis sobre las entidades que estarán presentes en la plataforma, dando lugar al [modelo Entidad-Relación](#).

Con los datos extraídos del diagrama Entidad-Relación, los requisitos y los casos de uso, se genera el [diagrama de flujo del sistema](#) y los [prototipos de las futuras pantallas](#) de la plataforma. El último punto de análisis y diseño mostrará la [estructura de la base de datos](#) del sistema.

En el siguiente punto se procede al [desarrollo de la aplicación](#) donde se podrá ver en detalle los componentes desarrollados, una presentación de las pantallas de la aplicación y su funcionamiento.

Más adelante, se muestra la sección de conclusiones, en la que se especifican las [conclusiones generales](#), posibles [trabajos futuros](#) y se indica el [presupuesto](#) de la realización del proyecto.

Finalmente se presenta el [glosario](#), que incluye una gran cantidad de conceptos que han aparecido a lo largo del documento y las [referencias](#) que han sido utilizadas como fuente de información.

El documento concluye con una serie de [anexos](#) que facilitan la comprensión del mismo y de la aplicación.

2. Estado de la cuestión

Actualmente existe un gran número de herramientas que permiten a los usuarios controlar y gestionar proyectos en desarrollo. Es por ello por lo que se hace necesario realizar un análisis a varias de las herramientas para comprobar qué funcionalidades ofrecen y qué carencias contienen.

Tal como se ha indicado en la sección [1.2 Objetivo](#), el principal objetivo es el análisis, diseño e implementación de una plataforma distribuida. Es decir que la plataforma donde se desarrollará el proyecto será una plataforma web. Por ello es necesario realizar un análisis de las tecnologías web existentes, comparando los pros y contras, para así seleccionar la tecnología que más se adecue al proyecto en desarrollo.

Este capítulo contiene dos secciones. La primera de ellas es el análisis de los sistemas de control de proyectos existentes en el mercado. La segunda sección es el análisis de las tecnologías web existentes.

2.1. Sistemas de Control de Proyectos

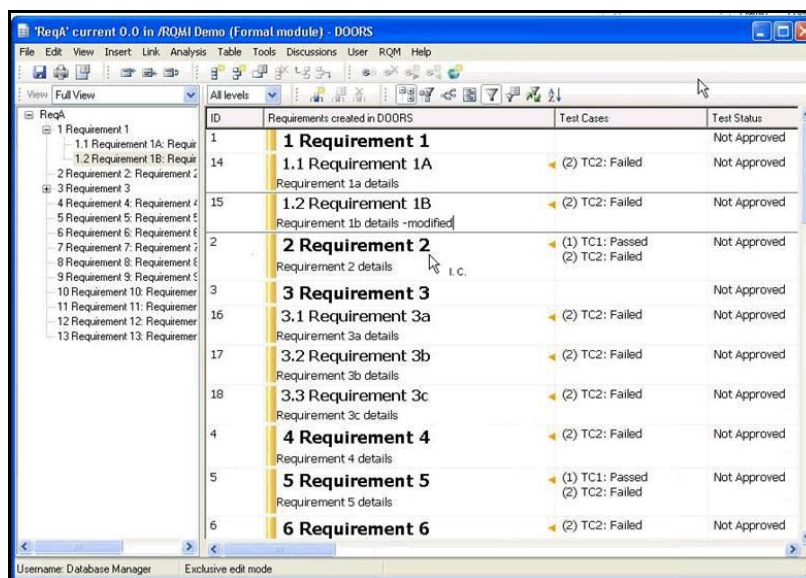
A lo largo de esta sección se expondrá la historia, características y todos aquellos elementos relevantes de los sistemas de control y gestión de proyectos que mantienen una similitud con el sistema a desarrollar.

2.1.1 IBM Rational Doors

A continuación se especifican las características principales y los puntos a mejorar del sistema [IBM Rational Doors](#).

Características principales

IBM Rational Doors es una aplicación de escritorio creada inicialmente por DOORS y posteriormente adquirida por IBM. Su principal objetivo es la optimización de la gestión de requisitos. Para cumplir este objetivo, la herramienta hace hincapié en la comunicación, ya que permite desarrollar requisitos más precisos y coherentes, lo que reduce la necesidad de reelaboración durante el proyecto.



ID	Requirements created in DOORS	Test Cases	Test Status
1	1 Requirement 1		Not Approved
14	1.1 Requirement 1A Requirement 1a details	(2) TC2: Failed	Not Approved
15	1.2 Requirement 1B Requirement 1b details -modified	(2) TC2: Failed	Not Approved
2	2 Requirement 2 Requirement 2 details	(1) TC1: Passed (2) TC2: Failed	Not Approved
3	3 Requirement 3		Not Approved
16	3.1 Requirement 3a Requirement 3a details	(2) TC2: Failed	Not Approved
17	3.2 Requirement 3b Requirement 3b details	(2) TC2: Failed	Not Approved
18	3.3 Requirement 3c Requirement 3c details	(2) TC2: Failed	Not Approved
4	4 Requirement 4 Requirement 4 details	(2) TC2: Failed	Not Approved
5	5 Requirement 5 Requirement 5 details	(1) TC1: Passed (2) TC2: Failed	Not Approved
6	6 Requirement 6	(2) TC2: Failed	Not Approved

Figura 1. Gestión de requisitos en IBM Rational Doors.

Rational Doors dispone de una interfaz de usuario con una amplia variedad de posibilidades. Las opciones más destacadas son:

- gestión de requisitos,
- control de cambios,
- visualización de requisitos distribuida,
- gestión de cartera de productos,
- sincronización de requisitos.

Desde un punto de vista más técnico, Rational Doors cuenta con una base de datos local y una base de datos on-line para sincronizar los requisitos entre diferentes usuarios. El programa cuenta con un editor de texto integrado, cuyo fin es eliminar la dependencia del usuario con otros programas de edición de texto.

Carencias y puntos a mejorar

IBM Rational Doors es un programa muy completo, pero su punto débil es la complejidad. Su uso se hace muy pesado ya que la interfaz de usuario es muy compleja.

Otros puntos a destacar son la falta de funcionalidades. Se centra mucho en la gestión de requisitos y su comunicación, echando en falta la gestión de proyectos, casos de uso, entregables, trazabilidad, planificación, etc.

Aunque el programa cuenta con una base de datos común para que los usuarios cuenten con datos sincronizados, Rational Doors es una aplicación de escritorio, por lo que es necesario tener el sistema instalado y configurado en cada ordenador. Dispone de un módulo web, pero es muy limitado: únicamente permite visualizar, modificar, aceptar o rechazar requisitos. Esta restricción hace que los usuarios únicamente puedan utilizar el programa si disponen de un sistema operativo [Microsoft Windows](#).

La idea de eliminar las dependencias de los usuarios con otros programas externos en cuanto a la edición de texto es buena, pero se queda en un mero intento, ya que el editor integrado es muy limitado.

Otra carencia es la falta de gestión de notificaciones frente a eventos importantes del proyecto en desarrollo. Dispone de un sistema de notificaciones opcional al realizar cambios en requisitos pero previamente la opción ha de ser configurada por cada usuario. El usuario debe disponer de una dirección de correo electrónico donde recibir la notificación correspondiente.

2.1.2 IBM Rational RequisitePro

A continuación se especifican las características principales y los puntos a mejorar del sistema [IBM Rational RequisitePro](#).

Características principales

Rational RequisitePro es un sistema de gestión de proyectos software creado por IBM. El sistema es muy similar a Rational Doors, aunque ofrece más posibilidades que su competidor. Su objetivo principal, al igual que Rational Doors, es la administración óptima de requisitos.

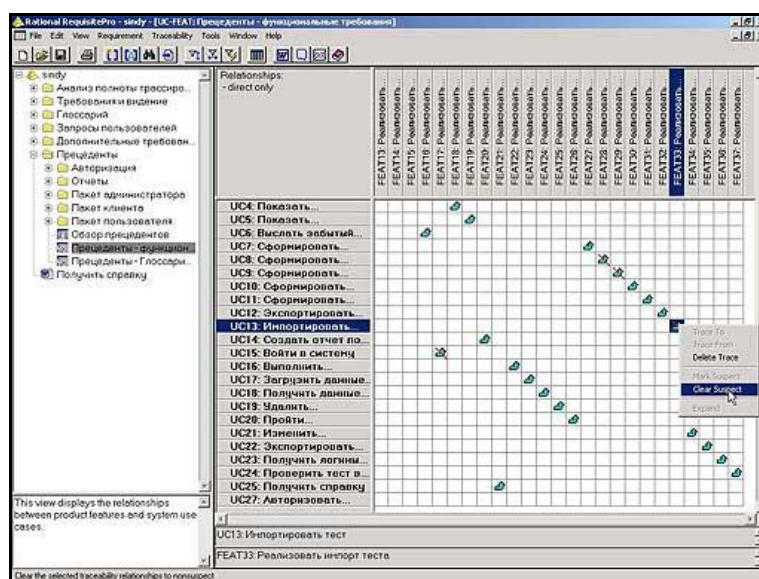


Figura 2. Matriz de trazabilidad en IBM Rational RequisitePro.

Rational RequisitePro es una aplicación de escritorio y sus funcionalidades principales son las siguientes:

- integración avanzada con Microsoft Word,
- gestión de requisitos,
- gestión de casos de uso,
- trazabilidad entre requisitos y casos de uso,
- acceso web para gestión distribuida,

- generación de informes,
- gestión del cambio,
- integración con otras herramientas de IBM.

Carencias y puntos a mejorar

Aunque Rational RequisitePro es más completo en cuanto a funcionalidades ofrecidas que Rational Doors, no consigue vencer a la barrera de la simplicidad. Su uso es muy complejo y el programa consume muchos recursos.

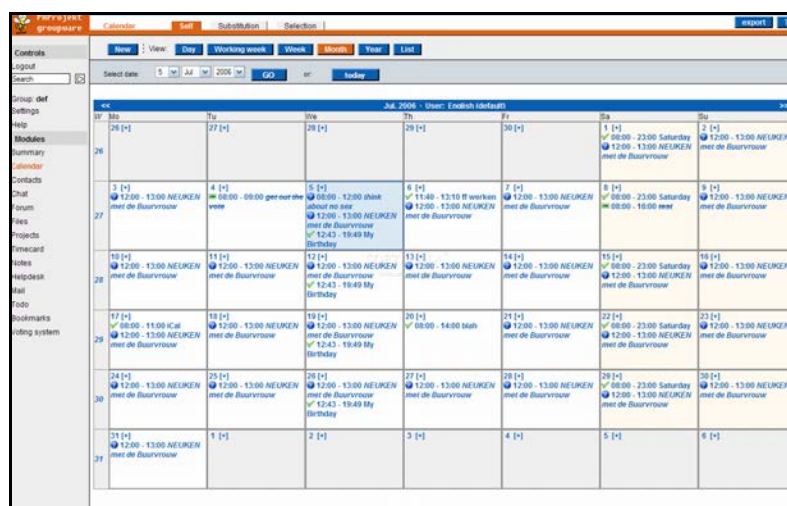
El sistema es una plataforma de escritorio, por lo que es necesario que esté instalada y configurada en un ordenador. Rational RequisitePro ofrece una aplicación web para evitar que otros usuarios tengan que instalar la aplicación de escritorio, pero ofrece funcionalidad limitada y peor aún, obliga al usuario principal a tener instalada la aplicación en un sistema con unos requisitos concretos para actuar como un servidor web. La aplicación debe estar instalada en un sistema operativo [Microsoft Windows 2000](#) como mínimo.

2.1.3 PHProjekt

A continuación se especifican las características principales y los puntos a mejorar del sistema [PHProjekt](#).

Características principales

PHProjekt es una aplicación web de [código abierto](#) de gestión de proyectos software. La aplicación, escrita en [PHP](#), fue creada a finales de 1990 por Alberto Günther y desde principios de 2006, el grupo Mayflower GmbH desarrolla y apoya PHProjekt.



Mo	Tu	We	Th	Fr	Sa	Su
26 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou	27 [1] 08:00 - 09:00 geenr de vete	28 [1] 08:00 - 12:00 (Mak diner no use 12:00 - 13:00 NEUKEN meet de Baurnvrou 12:43 - 19:49 My Birthday	29 [1] 11:40 - 13:10 W werks 12:00 - 13:00 NEUKEN meet de Baurnvrou	30 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou	1 [1] 08:00 - 23:00 Saturday 12:00 - 13:00 NEUKEN meet de Baurnvrou	2 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou
3 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou	4 [1] 08:00 - 09:00 geenr de vete	5 [1] 08:00 - 12:00 (Mak diner no use 12:00 - 13:00 NEUKEN meet de Baurnvrou 12:43 - 19:49 My Birthday	6 [1] 11:40 - 13:10 W werks 12:00 - 13:00 NEUKEN meet de Baurnvrou	7 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou	8 [1] 08:00 - 23:00 Saturday 12:00 - 13:00 NEUKEN meet de Baurnvrou	9 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou
10 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou	11 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou	12 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou 12:43 - 19:49 My Birthday	13 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou	14 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou	15 [1] 08:00 - 23:00 Saturday 12:00 - 13:00 NEUKEN meet de Baurnvrou	16 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou
17 [1] 08:00 - 11:00 (Mak 12:00 - 13:00 NEUKEN meet de Baurnvrou	18 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou	19 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou 12:43 - 19:49 My Birthday	20 [1] 08:00 - 16:00 Mak	21 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou	22 [1] 08:00 - 23:00 Saturday 12:00 - 13:00 NEUKEN meet de Baurnvrou	23 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou
24 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou	25 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou	26 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou 12:43 - 19:49 My Birthday	27 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou	28 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou	29 [1] 08:00 - 23:00 Saturday 12:00 - 13:00 NEUKEN meet de Baurnvrou	30 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou
31 [1] 12:00 - 13:00 NEUKEN meet de Baurnvrou	1 [1]	2 [1]	3 [1]	4 [1]	5 [1]	6 [1]

Figura 3. Gestión de planificación en PHProjekt.

Su característica principal es la gestión de proyectos de forma distribuida, de tal forma que el usuario pueda acceder a la aplicación desde cualquier sistema operativo con conexión a internet, sin tener que instalarla en el ordenador. Las funcionalidades que más destacan del sistema son las siguientes:

- gestión de proyectos,

- El sistema permite también la integración de Microsoft Outlook. Así los usuarios pueden acceder a sus correos electrónicos y sus calendarios a través de PHProjekt.

PHPProjekt está situado en el punto contrario a los sistemas de IBM analizados anteriormente. PHPProjekt es una plataforma muy ligera y su uso es muy sencillo, pero su punto débil es la limitación de funcionalidades referentes a la [gestión de proyectos](#). Aunque ofrece una gestión de proyectos básica, no brinda ninguna gestión de requisitos, casos de uso, productos u otros elementos relacionados con el desarrollo de software. Únicamente permite la creación de notas en [texto plano](#).

2.1.4 Microsoft Project

A continuación se especifican las características principales y los puntos a mejorar del sistema [Microsoft Project](#).

Microsoft Project es un software de administración de proyectos creado por Microsoft cuya función principal es la asistencia a administradores de proyectos para controlar diversos elementos que se encuentran en el desarrollo de un proyecto.

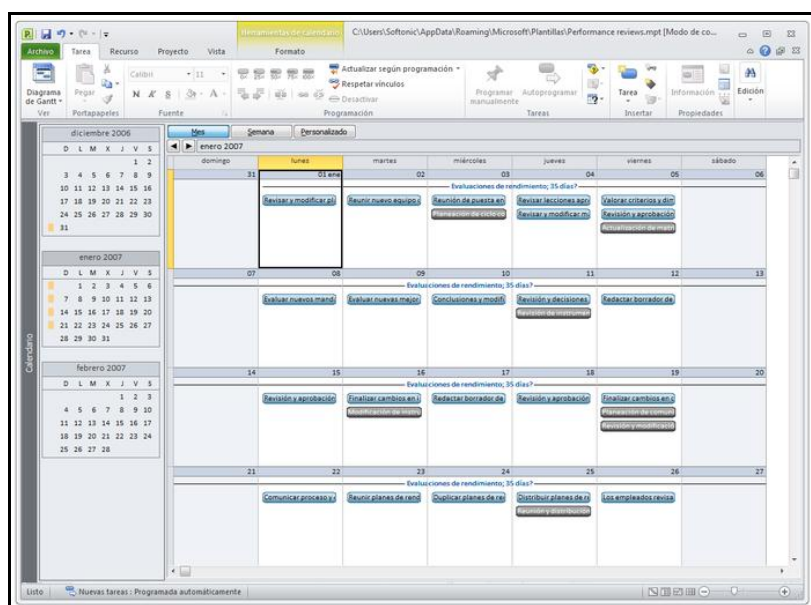


Figura 4. Gestión de planificación en Microsoft Project.

Las funcionalidades principales que ofrece el sistema son las siguientes:

- desarrollo de planes,
- asignación de recursos a tareas,
- seguimiento al progreso,
- administración del presupuesto,
- análisis de cargas de trabajo.

Microsoft Project se desarrolló por primera vez en 1984 y actualmente Microsoft continúa con su actualización y mantenimiento. Se trata de una aplicación de escritorio y está disponible para los sistemas operativos [Windows y Mac](#).

Actualmente la aplicación se incluye en el paquete [Microsoft Office](#), haciendo uso de ficheros locales del paquete para gestionar el almacenamiento de los datos. Gracias a ello, permite la integración de numerosas herramientas Office.

Carencias y puntos a mejorar

La principal carencia de Microsoft Project es el tipo de plataforma. Se trata de una aplicación cien por cien de escritorio, sin ninguna posibilidad de compartición de datos a través de web. Si un usuario desea utilizar la aplicación, debe tenerla instalada en su equipo con sistema operativo Windows o Mac.

La aplicación está orientada al control de proyectos llevada a cabo por un único usuario, ya que no ofrece ningún sistema de gestión de usuarios ni compartición de datos.

Aunque el uso de la aplicación es bastante sencillo, la funcionalidad ofrecida es muy limitada, ya que está destinada básicamente a la gestión de la planificación de un proyecto. No ofrece por tanto gestión de diferentes elementos como requisitos, casos de uso, entregables, trazabilidad, etc.

2.2. Tecnologías web

El proyecto a desarrollar será un sistema basado en web, por lo que es necesario realizar una comparación sobre las diferentes tecnologías web existentes.

Es muy posible que se obtuviera un resultado similar utilizando cada tecnología. Lo importante es analizar las diferentes facilidades que cada tecnología ofrece al desarrollador o al usuario final, los diferentes requisitos necesarios para implantar la tecnología o incluso el coste del hardware necesario para ofrecer un correcto funcionamiento.

En primer lugar se realizará un análisis a las diferentes tecnologías de desarrollo de aplicaciones web y en segundo lugar el análisis se centrará en los gestores de bases de datos relacionales.

2.2.1 Tecnologías de desarrollo de aplicaciones web

A continuación se muestra el análisis para las tecnologías JSP, PHP, ASP .NET, ColdFusion, Django, Ruby on Rails y C++.

JSP

JavaServer Pages ([JSP](#)) es una tecnología [Java](#) que permite generar contenido dinámico para web, en forma de documentos [HTML](#), [XML](#) o de otro tipo.

Java es un lenguaje orientado a objetos desarrollado por [Sun Microsystems](#). Este lenguaje es mitad compilado y mitad interpretado. La principal ventaja de este sistema es que es posible ejecutar aplicaciones Java en todos aquellos sistemas que dispongan de una Máquina Virtual Java ([JVM](#)). Además, muchas opciones de seguridad van implementadas en la propia JVM.



Las ventajas de JSP frente a otras tecnologías web son las siguientes:

- Es multiplataforma: Un mismo código Java es ejecutable, sin necesidad de modificarlo, en cualquier sistema operativo que disponga de una Máquina Virtual Java.
- Posee un “recolector de basura”: La JVM tiene un sistema integrado que se encarga de liberar la memoria que no se usa (variables y objetos en desuso, etc.).
- Se puede desarrollar a partir de frameworks: Java soporta gran cantidad de frameworks, tales como Struts, Jboss Seam, Tiles, etc., que facilitan la programación de aplicaciones y el uso de distintos patrones de diseño como el [MVC](#), [Singleton](#)...
- Posee un amplio soporte: Java cuenta con una gran comunidad de soporte.
- Es orientado a objetos: Java es un lenguaje totalmente orientado a objetos.

- Posibilita el uso de patrones de diseño: Como Java es un lenguaje que programación orientada a objetos es sencillo aplicar patrones de diseño para facilitar la reusabilidad del código.
- Permite encapsulación, herencia y polimorfismo: Estas técnicas facilitan el mantenimiento, reutilización y modularidad del producto final.
- Posee varios entornos de funcionamiento:
 - J2ME (Micro Edition): para dispositivos móviles, smartphones, etc.
 - J2SE (Standard Edition): para ordenadores de escritorio.
 - J2EE (Enterprise Edition): para aplicaciones empresariales.

Los requisitos para hacer funcionar un proyecto web en JSP son los siguientes:

- Necesario un servidor web que admita JSP, como por ejemplo Tomcat.
- Al tratarse de un lenguaje estructurado y modular, es necesario un entorno de desarrollo potente para facilitar el trabajo del desarrollador.
- Necesidad de librerías externas para lograr una conexión entre la aplicación y la base de datos.

PHP

[PHP](#) es un lenguaje de programación interpretado que se emplea principalmente en aplicaciones web de ejecución en servidor. PHP es software libre, por lo que está soportado por una gran comunidad de usuarios y no necesita una licencia privada para su uso.

Cuenta con una sintaxis muy parecida a otros lenguajes como [C++](#). Hoy por hoy es el lenguaje más utilizado para crear aplicaciones web junto con Java y el lenguaje [.NET](#) de Microsoft.



Las ventajas que hacen destacar a PHP sobre otros lenguajes son:

- Es multiplataforma: Se trata de un lenguaje interpretado por distintos servidores web como [Apache](#), [IIS](#), etc.
- Posee conexión con bases de datos: PHP tiene capacidad de conexión con la casi totalidad de los motores de bases de datos actuales.
- Es gratuito: Se presenta como una alternativa de fácil y rápido acceso, sin licencias ni costes extraordinarios.
- Se puede desarrollar a partir de frameworks: PHP tiene disponibles distintos frameworks [Open Source](#) que facilitan el desarrollo de aplicaciones. Algunos de estos frameworks son Kumbia, Symfony y CakePHP.
- No necesita un entorno de desarrollo complejo. Basta con un editor de texto para comenzar a desarrollar un proyecto web completo.

ASP .NET

[ASP.NET](#) es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es utilizado para construir sitios web dinámicos, aplicaciones web y servicios web [XML](#).

ASP.NET está construido sobre el [Common Language Runtime](#), permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por .NET Framework ([Python](#), [Visual Basic](#), [C#](#), etc.).



Los proyectos web creados mediante ASP .NET únicamente serán interpretados por servidores cuya plataforma sea Microsoft Windows.

Las ventajas que hacen destacar a ASP .NET sobre otros lenguajes son:

- Posibilidad de utilizar una gran variedad de lenguajes como C#, Visual Basic y [JScript](#) (cualquier lenguaje admitido por .NET Framework).
- Utiliza lenguajes orientados a aplicaciones muy potentes.
- Permite una gran estructuración del código y un desarrollo rápido gracias a los entornos de desarrollo de Microsoft.
- El sistema gestor de base de datos utilizado habitualmente es [Access](#), aunque mediante [ODBC](#) puede utilizar otros sistemas de bases de datos.
- Gran rendimiento gracias a que la aplicación se compila una sola vez al lenguaje nativo. En cada petición se produce una compilación Just In Time, es decir se compila desde el código nativo, lo que permite mucho mejor rendimiento. También permite el almacenamiento de caché en el servidor.

ColdFusion

ColdFusion es un lenguaje de programación creada por Allaire, empresa adquirida por Macromedia en 2002. El objetivo principal de ColdFusion es el acceso a bases de datos desde la web. Es una potente herramienta para realizar las funciones de acceso a la información alojada en bases de datos, utilización de programación personalizada, y presentación de la información utilizando formatos avanzados.

ColdFusion centra su potencialidad en la confiabilidad y el control de datos. Utiliza fuentes de datos [ODBC](#) que soporten sentencias [SQL](#).



Las ventajas principales de ColdFusion son las siguientes:

- Soporta cualquier sistema gestor de base de datos que soporte ODBC de 32 bits.
- Detecta errores producidos por el servidor SQL.
- Funciona perfectamente en una máquina remota. Su ejecución es óptima es servidores Microsoft Internet Information Server.
- Provee de ayuda que permite generar páginas [HTML](#) de forma dinámica.

- Permite la creación de estructuras condicionales dinámicamente para personalizar la solicitud de datos y el envío de los mismos hacia el cliente. Así mismo, diseña cadenas de datos para crear dinámicamente menús desplegables y para llenar listas de selección y listas de documentos.
- Dispone de soporte continuado por la empresa Macromedia.

Django

Django es un framework de desarrollo web de código abierto, escrito en [Python](#), que cumple en cierta medida el paradigma del [Modelo Vista Controlador](#). Fue desarrollado en primer lugar para gestionar varias páginas orientadas a noticias de la World Company de Lawrence, Kansas, y fue liberada al público bajo una licencia [BSD](#) en julio de 2005. El framework fue nombrado en alusión al guitarrista de jazz Django Reinhardt.

En junio de 2008 fue anunciado que la recién formada Django Software Foundation se haría cargo de Django en el futuro.



Las ventajas que ofrece Django son las siguientes:

- Proporciona una serie de características que facilitan el desarrollo rápido de páginas orientadas a contenidos.
- Proporciona una aplicación incorporada para administrar los contenidos, que puede incluirse como parte de cualquier página desarrollada con Django e incluso administrar varias páginas desarrolladas con Django a partir de una misma instalación.
- Proporciona un sistema incorporado de "vistas genéricas". Así el desarrollador no está obligado a escribir la lógica de ciertas tareas comunes.
- Cuenta con un sistema extensible de plantillas basado en etiquetas, con herencia de plantillas.

Ruby on Rails

Ruby es un lenguaje de programación interpretado, reflexivo y orientado a objetos. Fue creado por el programador japonés Yukihiro "Matz" Matsumoto, quien comenzó a trabajar en Ruby en 1993, y lo presentó públicamente en 1995.

Ruby combina una sintaxis inspirada en [Python](#) y [Perl](#) con características de programación orientada a objetos similares a [Smalltalk](#). Comparte también funcionalidad con el lenguaje de programación [Lisp](#). Ruby es un lenguaje de programación interpretado en una sola pasada y su implementación oficial es distribuida bajo una licencia de software libre.



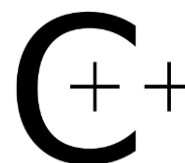
Ruby on Rails es un framework de aplicaciones web de código abierto escrito en el lenguaje Ruby, siguiendo el paradigma de la arquitectura [Modelo Vista Controlador](#). Trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones escribiendo menos código que con otros frameworks similares.

Las ventajas de Ruby frente a otros lenguajes son las siguientes:

- Ruby es un lenguaje orientado a objetos: todos los tipos de datos son un objeto, incluidas las clases y tipos que otros lenguajes definen como primitivas.
- Dispone de cuatro niveles de ámbito de variable: global, clase, instancia y local.
- Proporciona un recolector de basura automático.
- Las aplicaciones desarrolladas mediante Ruby son totalmente portables.
- Incluye carga dinámica de bibliotecas compartidas en la mayoría de las plataformas.

C++

C++ es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación [C](#) con mecanismos que permitieran la manipulación de objetos. Es por ello por lo que C++ es un lenguaje híbrido.



Posteriormente se añadieron facilidades de programación genérica, tales como la programación estructurada y la programación orientada a objetos. Por ello C++ es un lenguaje de programación multiparadigma.

Actualmente C++ es muy utilizado. Un ejemplo de ello es la red social [Facebook](#). La red social está desarrollada en PHP, pero debido al gran procesamiento que requerían los servidores de la aplicación, fue necesaria una transformación de PHP a C++. Para ello utilizaron un transformador de PHP a C++ llamado “[HipHop for PHP](#)”. Esto no quiere decir que Facebook esté desarrollado en C++. La aplicación continúa estando desarrollada en PHP pero gracias a HipHop se transforma el código fuente PHP a código C++ optimizado y éste es compilado utilizando g++.

Gracias a la transformación, Facebook ahorra un 50% de procesamiento, lo que conlleva a menos consumo de energía, menos servidores y por tanto menos costes. Este es un gran ejemplo de lo que se puede llegar a conseguir gracias a dos frameworks de alta calidad como son PHP y C++.

Las ventajas de C++ frente a otros lenguajes son las siguientes:

- Lenguaje de programación orientado a objetos y multiparadigma.
- Lenguaje muy potente y robusto destinado a la creación de sistemas complejos.
- Es un lenguaje muy empleado, por lo que hay disponible un gran número de fuentes de documentación para solucionar cualquier problema que pueda surgir durante el desarrollo.

Comparativa

Una vez descritas las alternativas se realiza una valoración de las mismas, considerando el impacto en el desarrollo del proyecto actual, tanto desde el punto de vista tecnológico como desde el punto de vista operativo.

Para cada tecnología web propuesta se valora el impacto en el proyecto y se establece su viabilidad económica. Para ello comprobará la existencia de varios aspectos tecnológicos y la posibilidad de los aspectos económicos.

A la hora de puntuar los aspectos tecnológicos se tendrán en cuenta las facilidades de desarrollo que ofrece la tecnología web así como los requisitos necesarios para lograr un sistema funcional, óptimo y de calidad pensando en futuros cambios.

El análisis comparativo se muestra en la siguiente tabla:

TECNOLOGÍA	Soporte AJAX	Soporte base de datos relacional	Soporte ORM	Independencia entorno de desarrollo	Multi-plataforma	Soporte de frameworks de migración de datos	Licencia	Coste
JSP	Sí	Sí	Sí	No	Sí	Mediante complementos	Propietaria	Bajo
PHP	Sí	Sí	Sí	Sí	Sí	Sí	Libre	Bajo
ASP .NET	Sí	Sí	Sí	No	No	Mediante complementos	Propietaria	Alto
ColdFusion	Sí	Sí	Sí	No	No	Mediante complementos	Propietaria	Medio
Django	Mediante complementos	Sí	Sí	No	Sí	Mediante complementos	Libre	Bajo
Ruby on Rails	Mediante complementos	Sí	Sí	No	Sí	Sí	Libre	Bajo
C++	Sí	Sí	Sí	Sí	Sí	No	Depende del compilador	Depende del compilador

Tabla 1. Comparativa de tecnologías web.

El lenguaje seleccionado es PHP. La razón de su puntuación y por tanto de su elección reside en los siguientes puntos:

- PHP no requiere entornos de desarrollo específicos en comparación con la mayor parte de las tecnologías analizadas. Aunque algunas de ellas, como JSP o Ruby, tampoco requieren un entorno de desarrollo específico, es recomendable utilizar entornos de desarrollo potentes para realizar un desarrollo acorde a la tecnología.
- PHP es un lenguaje libre, sin propietario. Esto hace que sea uno de los lenguajes más utilizados por los desarrolladores, por lo que existen numerosas fuentes de documentación y ayuda para prácticamente cualquier problema que pueda surgir durante la fase de implantación.
- Al tratarse de un lenguaje libre, no genera ningún coste. El único coste será la máquina servidora de la aplicación web. Dicho coste será más económico que si se hubiera escogido ASP .NET, ya que el servidor puede tener un sistema operativo libre, como Linux.
- PHP admite la utilización de MySQL de una forma rápida y sencilla. Esto será de gran utilidad para llevar a cabo la implementación de la base de datos diseñada en la sección “[Estructura de la base de datos](#)”.

- Aunque C++ también consigue una gran puntuación en la comparativa, este lenguaje está orientado a aplicaciones muy grandes con un gran consumo de recursos. La aplicación a desarrollar es de tamaño mediano, por lo que en principio se puede utilizar PHP sin problemas. Otro aspecto a destacar es la complejidad de C++, orientado a personas expertas. La curva de aprendizaje de C++ es mucho más exigente que en el caso de PHP.
- Se ha tenido también en cuenta el avance realizado por Facebook con el transformador [HipHop for PHP](#). Si en un futuro la aplicación consumiese muchos recursos, podría optarse por la optimización del código PHP a C++.

Se escoge por tanto PHP, ya que es una tecnología orientada a proyectos medianos que permite obtener resultados óptimos en un tiempo moderado.

2.2.2 Sistemas gestores de bases de datos

A continuación se muestra el análisis para los sistemas gestores de bases de datos MySQL, NoSQL, Oracle y PostgreSQL.

MySQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario. MySQL es software libre en un esquema de licenciamiento dual. Por un lado se ofrece bajo la [GNU GPL](#) para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar una licencia específica que les permita este uso.



Las ventajas de MySQL frente al resto de sistemas gestores de bases de datos son las siguientes:

- Amplio subconjunto del lenguaje SQL.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferente velocidad de operación, soporte físico, capacidad, distribución geográfica, transacciones, etc.
- Soporte para transacciones y claves ajenas.
- Proporciona una conectividad segura.
- Sistema muy rápido. Hasta 80 veces más que Oracle.
- Admite búsqueda e indexación de campos de texto.
- Permite escoger entre múltiples motores de almacenamiento para cada tabla.
- Soporte óptimo para la herramienta phpMyAdmin escrita en PHP. Facilidad de manejo con los datos almacenados gracias a la herramienta.

NoSQL

NoSQL hace referencia a una amplia clase de sistemas de gestión de bases de datos que difieren del modelo clásico del sistema de gestión de bases de datos relacionales. El aspecto más destacado es que no usan SQL como el principal lenguaje de consultas.

La mayoría de sistemas NoSQL emplean una arquitectura distribuida, manteniendo los datos de forma redundante en varios servidores, usando frecuentemente una [tabla hash](#) distribuida. De esta forma, el sistema puede realmente escalar añadiendo más servidores y el fallo en un servidor puede ser tolerado.



Un sistema de este tipo es MongoDB. Este sistema está orientado al almacenamiento de documentos en el que cada entrada o registro puede tener un esquema de datos diferente, con atributos o columnas que no tienen por qué repetirse de un registro a otro. Está escrito en C++, lo que le confiere cierta cercanía a los recursos de hardware de la máquina, de modo que es bastante rápido a la hora de ejecutar sus tareas.

Las ventajas de este tipo de sistemas son las siguientes:

- Pueden manejar gran cantidad de datos sin verse alterado su rendimiento.
- La gestión de datos no genera cuellos de botella.
- Los sistemas NoSQL son de código abierto.
- Disponen de una gran escalabilidad en el caso de ampliación de los sistemas.

Oracle

Oracle es un sistema de gestión de base de datos objeto-relacional desarrollado por Oracle Corporation. **ORACLE®** Recientemente, Oracle adquirió a Sun Microsystems y con ella la empresa encargada comercial de MySQL.

Oracle está considerado como uno de los sistemas de bases de datos más completos por su gran estabilidad y escalabilidad. La única edición gratuita es la Express Edition, que es compatible con las demás ediciones de Oracle Database.

Las ventajas de Oracle frente a otros sistemas son las siguientes:

- Oracle es el motor de base de datos relacional más usado a nivel mundial.
- Dispone de un lenguaje de diseño de bases de datos muy completo que permite implementar diseños activos, con [triggers](#) y procedimientos almacenados, con una integridad bastante potente.
- Permite el uso de particiones para mejorar la eficiencia de replicación e incluso ciertas versiones admiten la administración de bases de datos distribuidas.
- El software del servidor puede ejecutarse en multitud de sistemas operativos.
- Oracle está orientado a sistemas de datos de gran tamaño.

PostgreSQL

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es gestionado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma libre y desinteresada.



Las características de PostgreSQL más relevantes son las siguientes:

- Alta concurrencia. Mediante un sistema denominado MVCC (Acceso concurrente multiversión) PostgreSQL permite que mientras un proceso escriba en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.
- Amplia variedad de tipos nativos como textos con longitud ilimitada, figuras geométricas, números de precisión arbitraria o incluso tipos creados por el usuario.
- Dispone de otras características más comunes como [triggers](#), [claves ajenas](#), herencia de tablas y operaciones geométricas.

Comparativa

Al igual que con las tecnologías de desarrollo de aplicaciones web, se va a realizar una comparación entre los sistemas gestores de bases de datos analizados anteriormente.

Se tendrán en cuenta los aspectos técnicos, la viabilidad económica y la adecuación de cada uno de ellos al proyecto que se va a desarrollar con el lenguaje PHP.

En lo que se refiera a aspectos técnicos, se valorará la compatibilidad de cada sistema con los sistemas operativos más utilizados actualmente.

SISTEMA GESTOR	Windows	Linux	MacOS	Compatible con PHP	iOS	Android	Symbian	Viabilidad económica
MySQL	Sí	Sí	Sí	Sí	Mediante complementos	No	Sí	Alta
NoSQL (MongoDB)	Sí	Sí	Sí	Sí	No	No	No	Alta
Oracle	Sí	Sí	Sí	Mediante complementos	No	No	No	Baja
PostgreSQL	Sí	Sí	Sí	Sí	Sí	No	No	Alta

Tabla 2. Comparativa de sistemas gestores de bases de datos.

El sistema seleccionado es MySQL. La razón de su puntuación y por tanto de su elección reside en los siguientes puntos:

- MySQL es uno de los gestores de bases de datos con mejor rendimiento gracias a la velocidad en que realiza las operaciones.

- Para el uso particular es totalmente gratuito.
- No son necesarios muchos requerimientos para la elaboración de bases de datos. Esto deriva en un escaso consumo de recursos, por lo que el sistema puede ser ejecutado en una máquina con pocos recursos sin ningún problema.
- Unidos los dos puntos anteriores, se puede confirmar que MySQL es muy viable desde el punto de vista económico.
- MySQL es compatible con la mayoría de sistemas operativos existentes.

Se puede concluir que PHP unido a MySQL forman un sistema de desarrollo web muy ágil, de calidad y orientado a proyectos medianos.

Si a PHP y MySQL se le unen Linux como sistema operativo y Apache como servidor de aplicaciones, obtenemos el conocido modelo LAMP (Linux, Apache, MySQL y PHP).

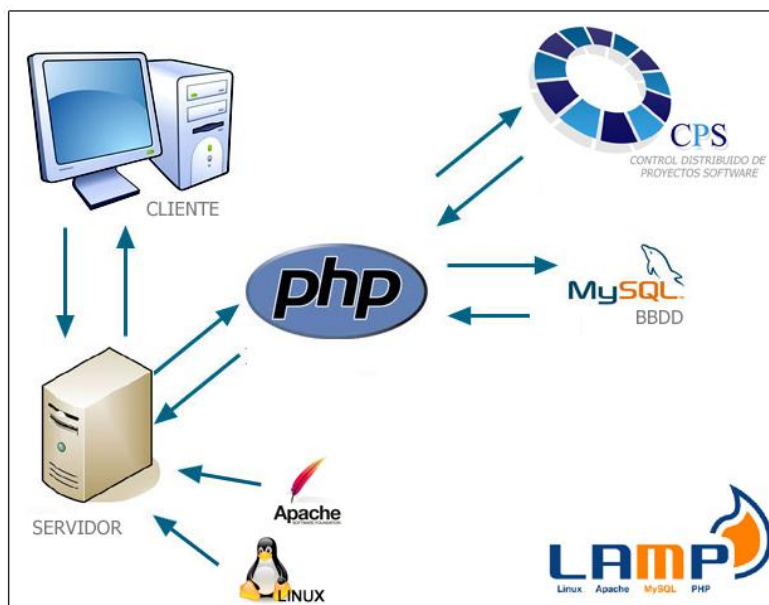


Figura 5. Modelo LAMP.

Este modelo proporciona una funcionalidad completa para un proyecto web como el que se desea desarrollar. Este es, por tanto, el modelo que se utilizará en el momento del desarrollo y la implementación del sistema.

El proyecto a desarrollar, al tratarse de una aplicación web, estará de forma adicional desarrollada en [HTML4](#) y [CSS3](#), las versiones de los lenguajes más compatibles con la mayor parte de navegadores de internet.

3. Análisis y Diseño

El análisis es la descomposición del problema que se desea resolver en partes más pequeñas que permitan su estudio de forma más sencilla. Posteriormente se procede al diseño, que consiste en la construcción de la arquitectura o plano conceptual, que permitirá dar lugar a la aplicación final a través del código.

3.1. Planificación del proyecto

Para el desarrollo de este proyecto es necesario establecer en primer lugar los elementos humanos y técnicos necesarios, reflejados en el [presupuesto](#), y el tiempo que será requerido para completarlo.

Para poder realizar el presupuesto es necesario conocer el tiempo que llevará el desarrollo del proyecto. Dependerá de dos aspectos: la complejidad de la aplicación a desarrollar, y la metodología usada.

Para la elección de la metodología es imprescindible conocer a priori la complejidad del proyecto sin entrar en detalle. Podemos asegurar que se trata de un proyecto sencillo, que involucra a un grupo reducido de desarrolladores. Las etapas principales del proyecto son dos:

- **Análisis y diseño:** Se describe y comprende correctamente el problema y las necesidades que requiere, y se busca la solución a estas necesidades. Mediante la toma de requisitos y los casos de uso se analiza el problema, y posteriormente con el diagrama de funciones se haya una solución. Esta es la labor de un analista de sistemas.
- **Implementación:** Siguiendo las instrucciones del analista, un programador debe escribir el código que genera la aplicación.

Aunque el sistema a desarrollar orientará a los proyectos a optar por un desarrollo guiado por la metodología [RUP](#), dicha metodología para el desarrollo del proyecto actual sería demasiado robusta y lenta. Otras metodología como ESA o Métrica v3 tienen el mismo problema. Se ha optado por tanto, por el uso de un híbrido entre una metodología ágil y la metodología RUP. Sus principales ventajas son las siguientes:

- Se enfatiza en el producto software, en vez de la documentación.
- Permite un desarrollo iterativo e incremental.
- Desarrollo dirigido por casos de uso.
- Se desarrolla únicamente lo necesario en el momento, no lo que se podría necesitar de cara a un futuro.
- Permite ver una versión temprana y funcional de la aplicación.

- Es ideal para proyectos y equipos pequeños y medianos, por el rápido desarrollo de software.

El desarrollo será iterativo e incremental, definiendo inicialmente las iteraciones, los casos de uso y los requisitos suplementarios. La funcionalidad definida por los casos de uso y los requisitos será desarrollada durante la iteración a la que hayan sido asignadas.

Al completar un número determinado de iteraciones se llegará a un lanzamiento de una nueva versión estable con las funcionalidades cubiertas hasta el momento según la planificación. Todas las iteraciones, casos de uso, requisitos y lanzamientos se describen en mayor detalle en las siguientes secciones.

A continuación se muestra la planificación que se llevará a cabo mediante un diagrama de Gantt. El diagrama ha sido generado con la herramienta [Microsoft Project](#) y se encuentra representado en la [Figura 6 de la siguiente página](#). Dicho diagrama se desglosará en detalle en la sección “[Plan de Iteraciones](#)”.

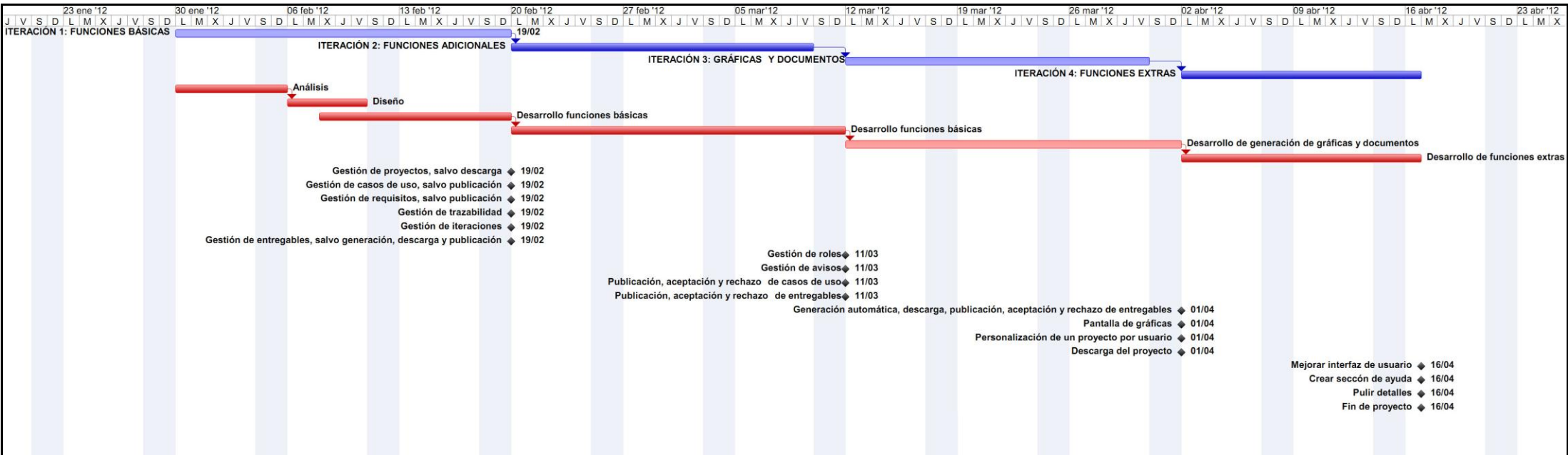


Figura 6. Diagrama de Gantt.

3.2. Plan de Iteraciones

Como resultado de la metodología seleccionada en la sección anterior, se procede a la elaboración del Plan de Iteraciones que guiará todo el desarrollo del proyecto. Dicho desarrollo se produce incrementalmente, introduciendo en cada iteración nuevas funcionalidades, y verificando su correcto funcionamiento.

En este caso, se va a dividir el trabajo en 4 iteraciones de 3 semanas cada una, salvo la última, que durará 2 semanas. De forma adicional, tras una o varias iteraciones se producen lanzamientos de una nueva versión funcional. Se ha elegido obtener 3 versiones: la primera muestra la funcionalidad básica del sistema, la segunda completa esa funcionalidad, y la tercera se encarga de los últimos detalles.

Iteraciones			
Nombre	Descripción	Fecha inicio	Fecha fin
Funciones extras	Mejorar interfaz, pulir detalles. Generación de la ayuda al usuario.	02/04/2012	16/04/2012
Gráficas y documentos	Gestión de gráficas, generación de documentos y descargas.	12/03/2012	01/04/2012
Funciones adicionales	Completar condiciones, acciones, interacciones y otras funciones.	20/02/2012	11/03/2012
Funciones básicas	Funcionalidades básicas que permiten hacer uso de la aplicación.	30/01/2012	19/02/2012

Tabla 3. Plan de iteraciones.

Funciones básicas			
Descripción:	Funcionalidades básicas que permiten hacer uso de la aplicación:		
	<ul style="list-style-type: none"> • Gestión de proyectos, salvo descarga. • Gestión de casos de uso, salvo publicación. • Gestión de trazabilidad. • Gestión de requisitos, salvo publicación. • Gestión de entregables, salvo generación automática, descarga y publicación. • Gestión de iteraciones. 		
Fecha inicio:	30/01/2012 12:00:00 AM Europa/Madrid	Fecha fin:	19/02/2012 11:59:59 PM Europa/Madrid

Tabla 4. Iteración 1: Funciones básicas.

Funciones adicionales			
Descripción:	Completar condiciones, acciones, interacciones y otras funciones:		
	<ul style="list-style-type: none"> • Gestión de roles por usuario. • Gestión de avisos. • Publicación, aceptación y rechazo de casos de uso. • Publicación, aceptación y rechazo de requisitos. 		
Fecha inicio:	20/02/2012 12:00:00 AM Europa/Madrid	Fecha fin:	11/03/2012 11:59:59 PM Europa/Madrid

Tabla 5. Iteración 2: Funciones adicionales.

Gráficas y documentos			
Descripción:	Completar funcionalidades:		
	<ul style="list-style-type: none"> • Generación automática, descarga, publicación, aceptación y rechazo de entregables. • Pantalla de gráficas. • Personalización de gráficas. • Descarga del proyecto. 		
Fecha inicio:	12/03/2012 12:00:00 AM Europa/Madrid	Fecha fin:	01/04/2012 11:59:59 PM Europa/Madrid

Tabla 6. Iteración 3: Gráficas y documentos.

Funciones extras			
Descripción:	Añadir otras funcionalidades:		
	<ul style="list-style-type: none"> • Mejorar interfaz de usuario. • Crear sección de ayuda. • Pulir detalles. 		
Fecha inicio:	02/04/2012 12:00:00 AM Europa/Madrid	Fecha fin:	16/04/2012 11:59:59 PM Europa/Madrid

Tabla 7. Iteración 4: Funciones extra.

3.3. Casos de uso

A continuación se muestra de una forma gráfica todas las funcionalidades generales que abarcará la aplicación.

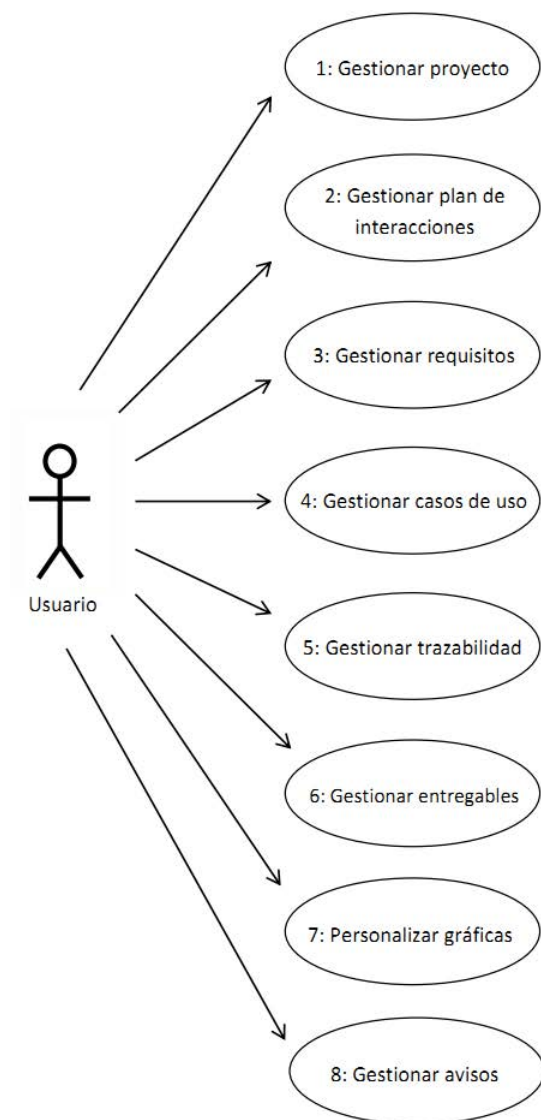


Figura 7. Funcionalidad general del sistema.

A continuación se muestra cada caso de uso desglosado. Hay que destacar que aunque el futuro sistema contará con diferentes tipos de usuario, los casos de uso tratarán a los usuarios como un único usuario. Los diferentes tipos de usuario y las acciones que podrán llevar a cabo están disponibles en la sección [3.4 Requisitos suplementarios](#).

CU1: Gestionar proyectos

Los usuarios del sistema podrán gestionar diferentes proyectos desde la aplicación. Los proyectos dispondrán de un ciclo de vida, desde su creación hasta su finalización y descarga.

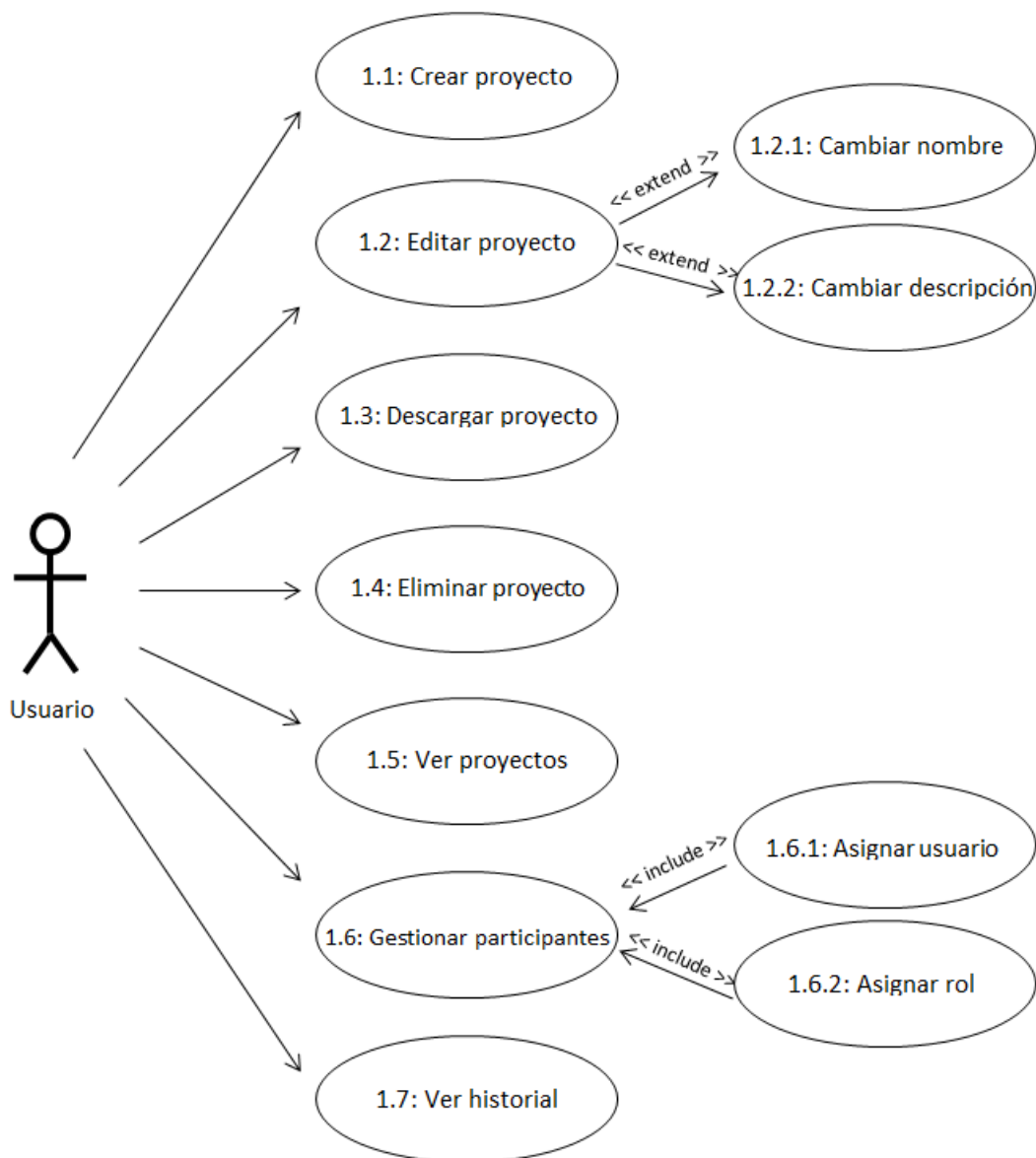


Figura 8. CU1: Gestionar proyectos.

A continuación se muestra en detalle cada funcionalidad relacionada con la gestión de los proyectos.

CU1.1 Crear proyecto

El usuario podrá crear un nuevo proyecto desde la futura aplicación siempre y cuando disponga de los permisos suficientes (Sección [3.4 Requisitos suplementarios](#)). Para crear el proyecto, el usuario deberá incluir un título y una descripción.

Tras la creación del nuevo proyecto, el usuario podrá llevar a cabo las funcionalidades que se definen en los siguientes casos de uso.

CU1.2 Editar proyecto

El usuario podrá editar la información del proyecto que fue indicada en su creación o anterior edición.

CU1.2.1 Cambiar nombre

El usuario podrá modificar el nombre o título de un proyecto creado.

CU1.2.2 Cambiar descripción

El usuario podrá modificar la descripción de un proyecto creado.

CU1.3 Descargar proyecto

El usuario podrá obtener un fichero .zip cuyo contenido serán todos los entregables generados tras el desarrollo del proyecto ([Sección CU6 Gestionar entregables](#)).

CU1.4 Eliminar proyecto

El usuario podrá eliminar un proyecto creado. La eliminación significará un borrado permanente del proyecto así como todos los elementos que pertenezcan al mismo.

CU1.5 Ver proyectos

El usuario dispondrá de una tabla en la que podrá ver los proyectos a los que pertenece junto a los detalles más relevantes.

CU1.6 Gestionar participantes

El usuario podrá seleccionar qué usuarios pertenecen a un proyecto y con qué roles.

CU1.6.1 Asignar usuario

El usuario podrá asignar a otros usuarios a un proyecto, siempre que cuente con los permisos necesarios (Sección [3.4 Requisitos suplementarios](#)). De igual forma, podrá desasignar usuarios.

CU1.6.2 Asignar rol

El usuario podrá establecer un rol específico a un usuario en un proyecto en concreto. Asimismo, el usuario podrá cambiar un rol anterior y asignar un rol nuevo.

CU1.7 Ver historial

El usuario tendrá un control absoluto sobre las acciones que se realicen en el proyecto. Toda acción quedará registrada en el historial, indicando quién lo ha realizado, el qué ha realizado, qué valor ha sufrido la acción realizada y la fecha en la que se realizó. El usuario podrá ordenar el historial de forma ascendente o descendente, según convenga.

CU2: Gestionar plan de iteraciones

El sistema orientará de forma opcional el desarrollo de proyectos mediante la metodología RUP.

Una de las características principales de la metodología RUP es que proporciona un desarrollo iterativo e incremental. Esto quiere decir que el desarrollo del proyecto se realiza en iteraciones. Para cada iteración o ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un mini proyecto cuyo núcleo fundamental está constituido

por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo. En concreto RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié dependiendo de las exigencias del proyecto.

Las fases en las que RUP divide el proyecto y por tanto las que el sistema proporcionará de forma opcional son: Inicio, Elaboración, Construcción y Transición. La distribución del trabajo en cada una de las fases dependerá del proyecto a realizar por lo que el usuario no estará obligado a realizar productos específicos para cada fase. Dichos productos serán los indicados por el usuario al crear cada iteración.



Figura 9. CU2: Gestionar plan de iteraciones.

Resumiendo los elementos de un Plan de Iteraciones, cada una de las fases se compone de diferentes iteraciones creadas por el usuario. En cada iteración se indica un

producto o productos a obtener y de forma opcional, un responsable de la realización de cada producto.

CU2.1 Crear iteración

El usuario podrá crear una iteración para una fase en concreto. El usuario indicará también el nombre de la iteración y la duración en semanas de la misma. El siguiente paso es la asignación de productos.

CU2.1.1 Añadir producto

El usuario podrá indicar qué productos se obtendrán a lo largo de la iteración para así pasar a la siguiente iteración del Plan. La selección de productos será opcional ya que el usuario podrá necesitar otro producto que no esté listado. Este caso se explica en más detalle en la sección [3.4 Requisitos suplementarios](#).

CU2.1.1.1 Asignar responsable

El usuario podrá indicar de forma opcional, qué usuario es el responsable de un producto.

CU2.2 Editar iteración

El usuario podrá modificar las diferentes propiedades de la iteración que fueron indicados en el momento de su creación.

CU2.2.1 Cambiar nombre

El usuario podrá cambiar el nombre de la iteración.

CU2.2.2 Cambiar duración

El usuario podrá modificar la duración en semanas de la iteración.

CU2.2.3 Eliminar producto

El usuario podrá eliminar productos que se obtendrían al finalizar la iteración.

CU2.2.4 Cambiar responsables

El usuario podrá modificar los responsables por cada producto de cada iteración.

CU2.3 Eliminar iteración

El usuario podrá eliminar la iteración.

CU2.4 Ver iteraciones

El usuario dispondrá de una tabla con las iteraciones creadas en un proyecto en concreto. Dichas iteraciones estarán clasificadas dependiendo de la fase a la que pertenezcan.

Por cada iteración, desde la propia tabla, el usuario podrá llevar a cabo diferentes acciones:

- Ver los productos de cada iteración ([CU2.6 Listar productos](#)).
- Editar iteración ([CU2.2 Editar iteración](#)).
- Ver el historial de la iteración ([CU2.5 Ver historial](#)).

- Eliminar iteración ([CU2.3 Eliminar iteración](#)).

CU2.5 Ver historial

El usuario podrá observar todas las acciones que se han realizado sobre la iteración seleccionada. Podrá observar también el responsable de cada acción realizada.

CU2.6 Listar productos

El usuario podrá observar los productos que se esperan obtener durante el desarrollo de la iteración seleccionada.

CU2.7 Pasar de fase

El usuario podrá pasar de fase en un proyecto concreto siempre que exista una iteración creada en la fase siguiente y se hayan generado todos los productos de la iteración actual.

CU2.8 Pasar de iteración

El usuario podrá pasar de iteración en un proyecto concreto siempre que exista una iteración siguiente y se hayan generado todos los productos de la iteración actual.

CU3: Gestionar requisitos

Un proyecto software se crea para servir a sus usuarios. Por lo tanto, para construir un sistema de calidad se debe conocer qué es lo que los usuarios quieren y necesitan. Hay que destacar que el término "usuarios" no se refiere únicamente a los usuarios humanos, sino a otros sistemas. En este contexto, el término "usuario" representa algo o alguien que interactúa con el sistema a desarrollar.

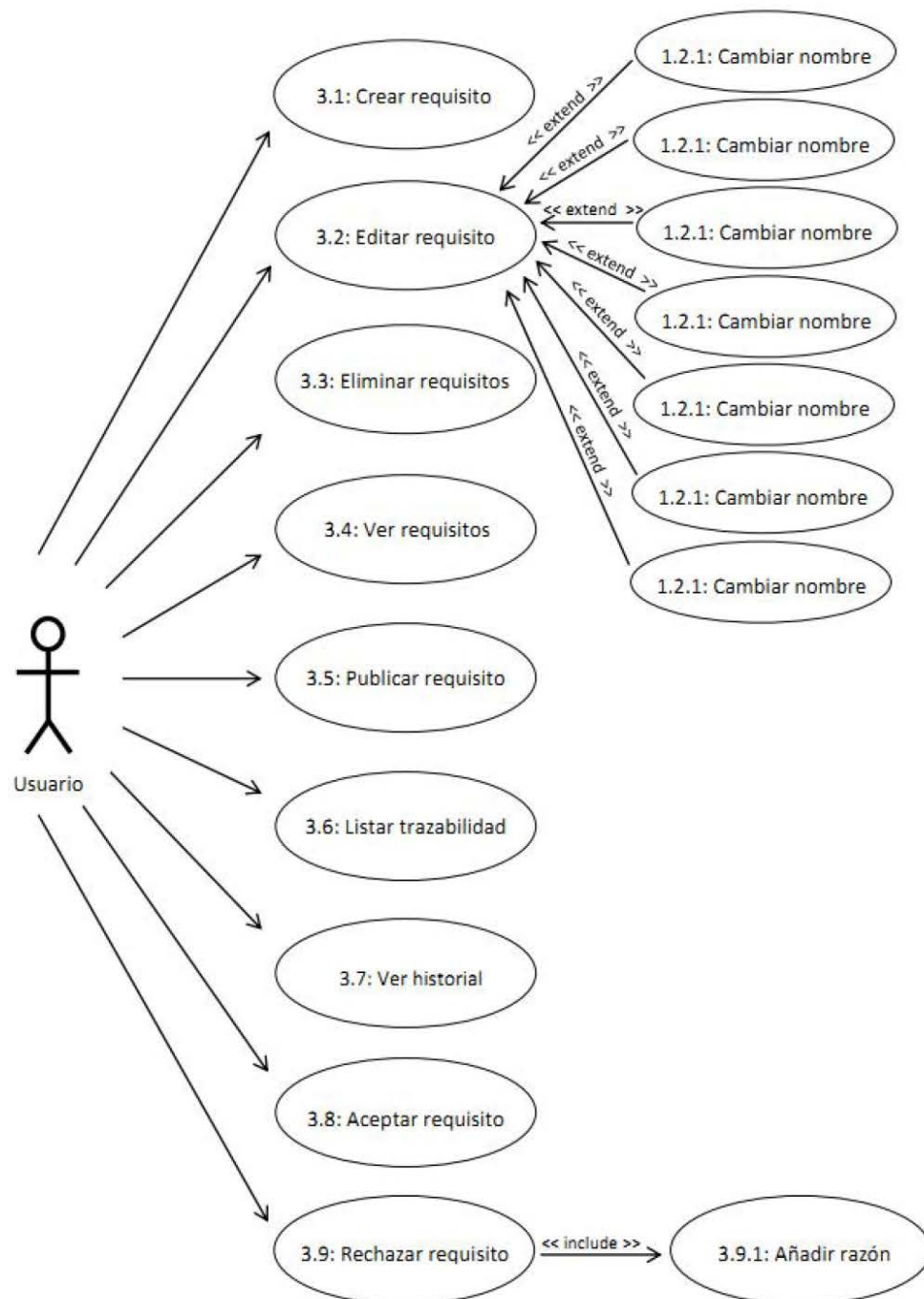


Figura 10. CU3: Gestionar requisitos.

En la metodología [RUP](#), un caso de uso es una pieza en la funcionalidad del sistema que le da al usuario un resultado de valor. Los casos de uso capturan los requisitos funcionales. El usuario podrá gestionar tanto requisitos funcionales como no funcionales.

CU3.1 Crear requisito

El usuario podrá crear un requisito. Para ello deberá incluir la información del tipo de trazabilidad, el título, la descripción, el impacto, el esfuerzo, el tipo de requisito y el subtipo. En la sección [3.4 Requisitos suplementarios](#) se indican todos los detalles de cada propiedad del requisito.

CU3.2 Editar requisito

El usuario podrá editar información de cada requisito ya creado.

CU3.2.1 Cambiar trazabilidad

El usuario podrá modificar el tipo de trazabilidad del requisito.

CU3.2.2 Cambiar título

El usuario podrá modificar el título del requisito.

CU3.2.3 Cambiar descripción

El usuario podrá modificar la descripción del requisito.

CU3.2.4 Cambiar tipo

El usuario podrá modificar el tipo de requisito.

CU3.2.5 Cambiar subtipo

El usuario podrá modificar el subtipo del requisito.

CU3.2.6 Cambiar impacto

El usuario podrá modificar el impacto del requisito.

CU3.2.7 Cambiar esfuerzo

El usuario podrá modificar el esfuerzo del requisito.

CU3.3 Eliminar requisito

El usuario podrá eliminar permanentemente el requisito.

CU3.4 Ver requisitos

El usuario podrá ver un listado con todos los requisitos creados. El listado de los requisitos mostrará toda la información de cada requisito creado. Además el usuario dispondrá de una serie de acciones para realizar sobre el requisito seleccionado:

- Gestionar trazabilidad ([CU3.6 Listar trazabilidad](#)).
- Editar ([CU3.2 Editar requisito](#)).
- Ver historial ([CU3.7 Ver historial](#)).
- Publicar ([CU3.5 Publicar requisito](#)).
- Eliminar ([CU3.3 Eliminar requisito](#)).

- Aceptar ([CU3.8 Aceptar requisito](#)).
- Rechazar ([CU3.9 Rechazar requisito](#)).

CU3.5 Publicar requisito

El usuario podrá publicar el requisito para que sea evaluado por el Project Manager o directamente por el cliente, dependiendo del rol del usuario.

CU3.6 Listar trazabilidad

Si el requisito dispone de trazabilidad con algún caso de uso, el usuario podrá observar el listado de dichas dependencias.

CU3.7 Ver historial

El usuario podrá ver todas las acciones que se han realizado sobre el requisito.

CU3.8 Aceptar requisito

El usuario podrá aceptar el requisito.

CU3.9 Rechazar requisito

El usuario podrá rechazar el requisito.

CU3.9.1 Añadir razón

De forma opcional, el usuario podrá indicar el motivo del rechazo.

CU4: Gestionar casos de uso

En la metodología [RUP](#), un caso de uso es una pieza en la funcionalidad del sistema que le da al usuario un resultado de valor. Todo el conjunto de casos de uso constituyen el modelo de casos de uso el cual describe la funcionalidad completa del sistema.

Este modelo reemplaza la tradicional especificación funcional de un sistema. Una especificación funcional tradicional se concentra en responder a la pregunta: ¿Qué debe hacer el sistema? La estrategia de casos de uso responde a una pregunta aún más específica: ¿Qué debe hacer el sistema por cada usuario? Esta estrategia fuerza a pensar en términos de valor a los usuarios y no únicamente en términos de las funciones que se relacionen con cada usuario.

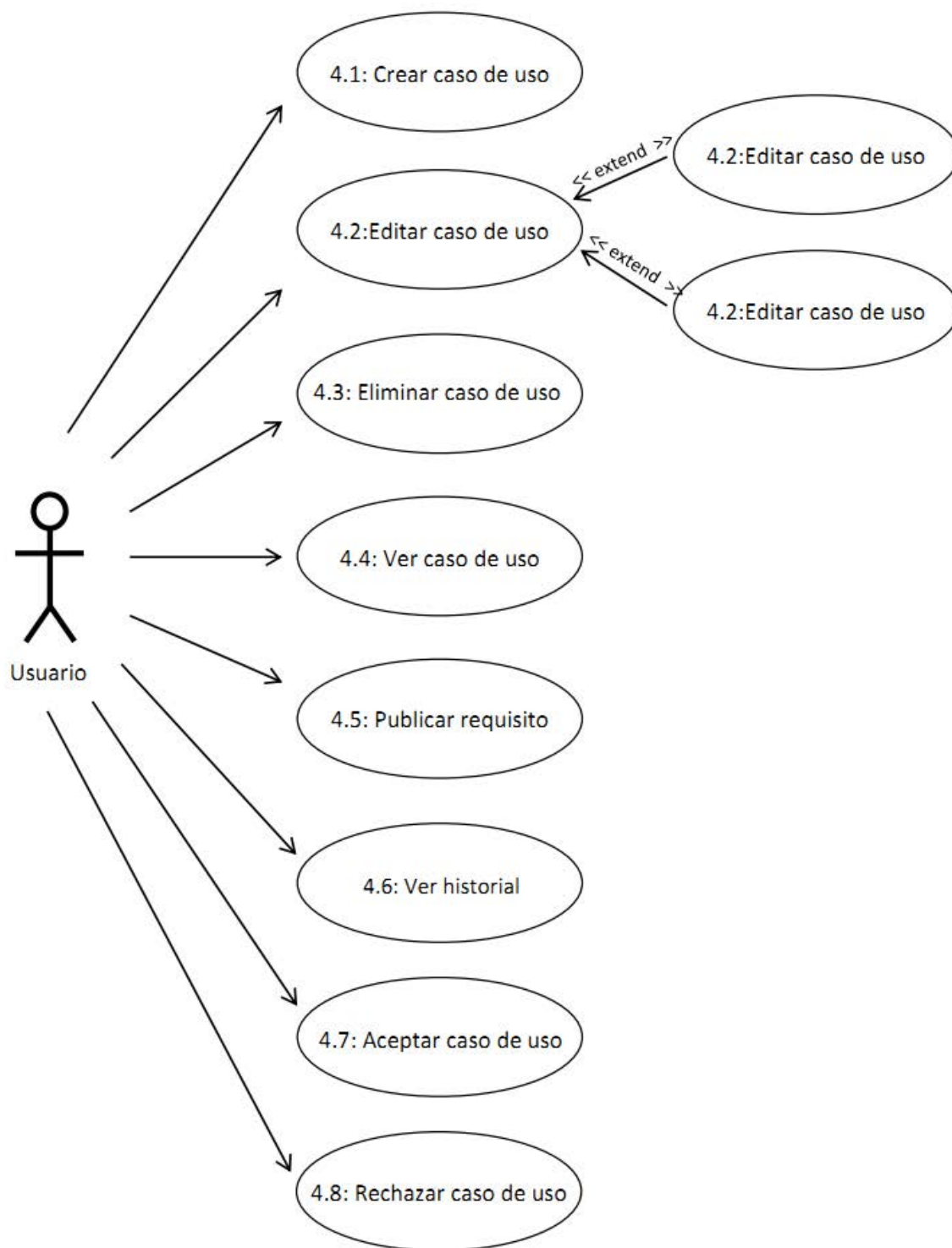


Figura 11. CU4: Gestionar casos de uso.

Los casos de uso no son solamente una herramienta para especificar los requisitos del sistema, también dirigen su diseño, implementación y pruebas. Se puede afirmar por tanto que los casos de uso, a través de los requisitos entre otros elementos, dirigen el proceso de desarrollo.

CU4.1 Crear caso de uso

El usuario podrá crear un nuevo caso de uso indicando un título descriptivo y la imagen que contenga el esquema del caso de uso.

CU4.2 Editar caso de uso

El usuario podrá modificar la información indicada en el momento de la creación del caso de uso.

CU4.2.1 Cambiar título

El usuario podrá modificar el título del caso de uso.

CU4.2.2 Cambiar imagen

El usuario podrá cambiar la imagen del caso de uso.

CU4.3 Eliminar caso de uso

El usuario podrá eliminar de forma permanente el caso de uso.

CU4.4 Ver casos de uso

El usuario podrá ver el listado de los casos de uso, que mostrará por cada fila toda la información de cada caso de uso creado. Además dispondrá de una serie de acciones a realizar sobre el caso de uso seleccionado:

- Editar ([CU4.2 Editar caso de uso](#)).
- Ver historial ([CU4.6 Ver historial](#)).
- Publicar ([CU4.5 Publicar caso de uso](#)).
- Eliminar ([CU4.3 Eliminar caso de uso](#)).
- Aceptar ([CU4.7 Aceptar caso de uso](#)).
- Rechazar ([CU4.8 Rechazar caso de uso](#)).

CU4.5 Publicar caso de uso

El usuario podrá publicar el caso de uso para que sea evaluado por el Project Manager o directamente por el cliente, dependiendo del rol del usuario.

CU4.6 Ver historial

El usuario podrá ver todas las acciones que se han realizado sobre el caso de uso.

CU4.7 Aceptar caso de uso

El usuario podrá aceptar el caso de uso.

CU4.8 Rechazar caso de uso

El usuario podrá rechazar el caso de uso.

CU4.8.1 Añadir razón

De forma opcional, el usuario podrá indicar el motivo del rechazo.

CU5: Gestionar trazabilidad

Si en un proyecto existen requisitos cuyo tipo de trazabilidad está definido mediante la dependencia con un caso de uso, el usuario deberá gestionar dichas dependencias entre los requisitos y casos de uso haciendo uso de una matriz de trazabilidad.

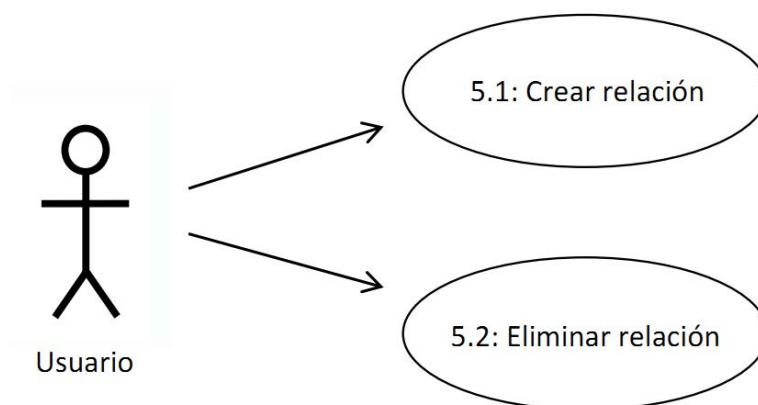


Figura 12. CU5: Gestionar trazabilidad.

El usuario podrá realizar las siguientes tareas para gestionar la matriz de trazabilidad de un proyecto.

CU5.1 Crear relación

El usuario podrá crear una relación requisito-caso de uso, que significará que el requisito depende del caso de uso seleccionado, y viceversa.

CU5.2 Eliminar relación

El usuario podrá eliminar una relación requisito-caso de uso, que significará que el requisito no depende del caso de uso seleccionado, y viceversa.

CU6: Gestionar entregables

Un entregable es un producto que se desea obtener de forma parcial o total al finalizar una iteración o una fase. La metodología [RUP](#) (Rational Unified Process) se refiere a estos productos mediante el término "artefactos".

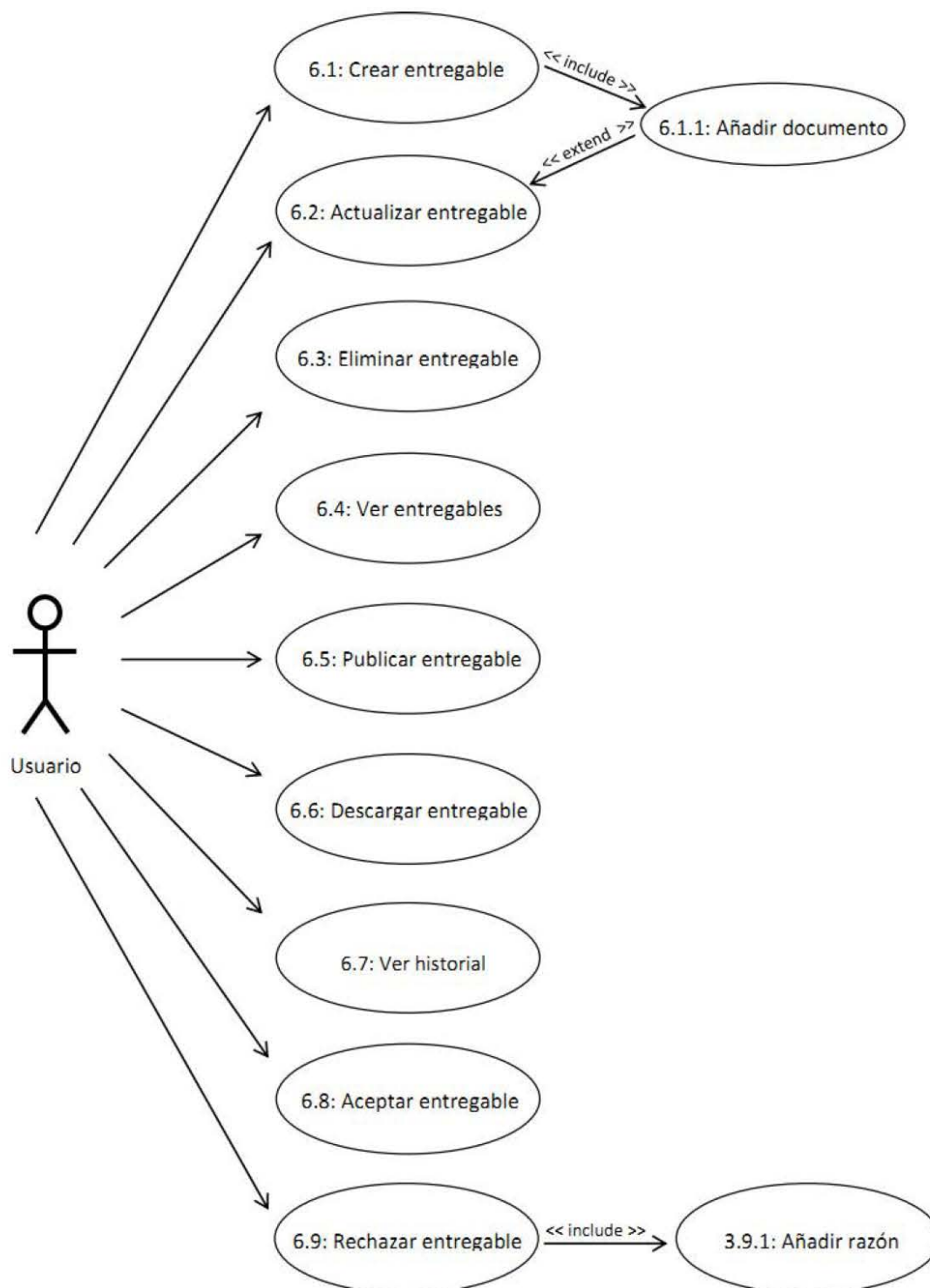


Figura 13. CU6: Gestionar entregables.

RUP define los artefactos e incluso los clasifica por las fases en los que se obtienen. El sistema a desarrollar no será tan estricto, ya que permitirá realizar un plan de iteraciones

abierto, dando la posibilidad al usuario de seleccionar los productos que se esperan obtener o incluso generar productos no contemplados por [RUP](#).

CU6.1 Crear entregable

El usuario podrá crear un nuevo entregable. Para ello dispondrá de una lista con los diferentes tipos de entregables que podrá generar. Hay que destacar que el listado de productos que se muestran son los correspondientes a la iteración y fase actuales. Si el usuario necesitase generar un producto diferente a los listados, podrá tomar dos caminos:

- Modificar el Plan de Iteraciones: el usuario deberá añadir el producto como objetivo de la iteración actual ([CU2.2 Editar iteración](#)).
- Crear "Otro entregable". El usuario tendrá disponible la opción "Otro". Si selecciona esta opción, el usuario podrá indicar el título y la descripción del nuevo entregable.

CU6.1.1 Añadir documento

El usuario tendrá disponible un apartado para la carga de ficheros externos al sistema. Si el entregable seleccionado no pudiese ser generado de forma automática, sería obligatorio incluir documentos externos.

CU6.2 Actualizar entregable

El usuario podrá crear una nueva versión del entregable, de tal forma que contendrá información actualizada del proyecto u otra información externa.

Existirán dos tipos de actualización: la actualización de documentos que generen contenido de forma automática y la actualización de documentos que dependan por completo de la información externa que el usuario ha cargado.

CU6.3 Eliminar entregable

El usuario podrá eliminar permanentemente el entregable.

CU6.4 Ver entregables

El usuario podrá observar el listado con todos los entregables creados. El listado de los entregables mostrará por cada fila toda la información de cada entregable creado. Además dispone de una serie de acciones para realizar sobre el entregable seleccionado:

- Actualizar ([CU6.2 Actualizar entregable](#)).
- Publicar ([CU6.5 Publicar entregable](#)).
- Descargar ([CU6.6 Descargar entregable](#)).
- Eliminar ([CU6.3 Eliminar entregable](#)).
- Aceptar ([CU6.7 Aceptar entregable](#)).
- Rechazar ([CU6.8 Rechazar entregable](#)).

CU6.5 Publicar entregable

El usuario podrá publicar el caso de uso para que sea evaluado por el Project Manager o directamente por el cliente, dependiendo del rol del usuario.

CU6.6 Descargar entregable

El usuario podrá descargarse el entregable para almacenarlo en su ordenador.

CU6.7 Aceptar entregable

El usuario podrá aceptar el entregable.

CU6.8 Rechazar entregable

El usuario podrá rechazar el entregable.

CU6.8.1 Añadir razón

De forma opcional, el usuario podrá indicar el motivo del rechazo.

CU7: Personalizar gráficas

El usuario podrá personalizar las gráficas que se mostrarán en la pantalla principal de cada proyecto al que esté asignado. Esto es de gran utilidad ya que cada usuario puede centrarse en las áreas que correspondan a su rol y tareas.

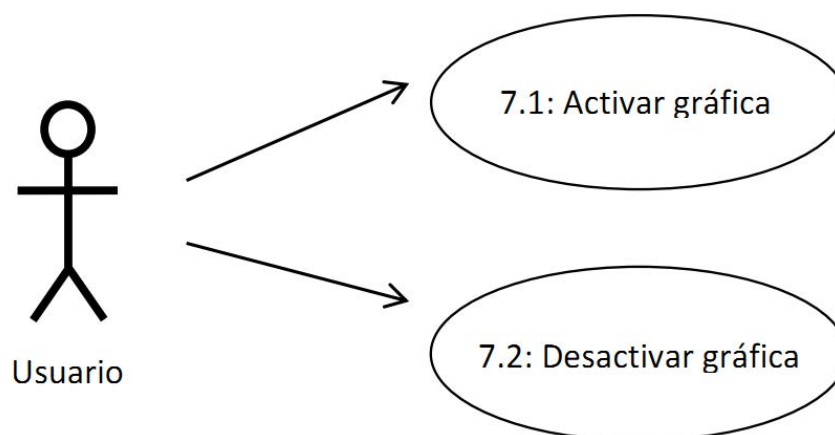


Figura 14. CU7: Personalizar gráficas.

Los diferentes tipos de gráficas están especificados en la sección [3.4 Requisitos suplementarios](#).

CU7.1 Activar gráfica

El usuario podrá activar una gráfica específica por cada proyecto al que pertenezca.

CU7.2 Desactivar gráfica

El usuario podrá desactivar una gráfica específica por cada proyecto al que pertenezca.

CU8: Gestionar avisos

El futuro sistema dispondrá de un módulo de avisos desde el que los usuarios de un mismo proyecto puedan recibir notificaciones o comunicarse entre ellos.

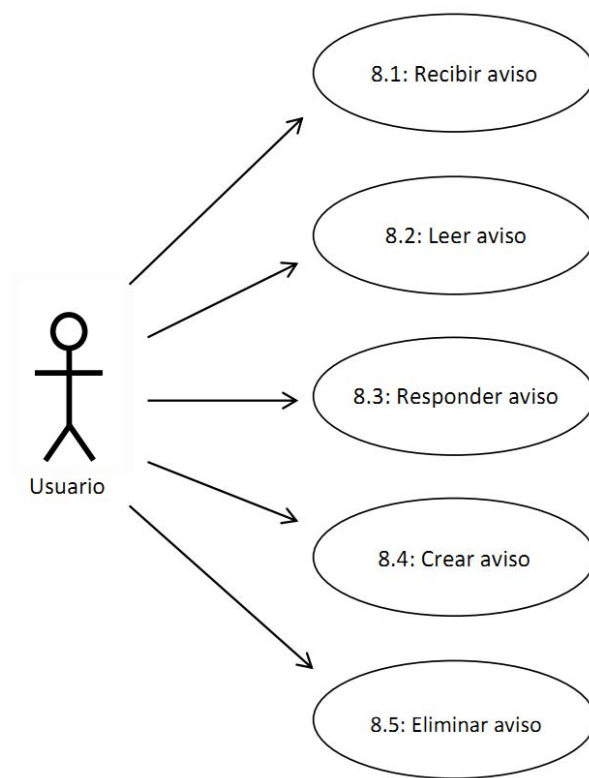


Figura 15. CU8: Gestionar avisos.

El usuario podrá crear nuevos avisos, así como responder o eliminar avisos ya existentes.

CU8.1 Recibir aviso

El usuario recibirá avisos automáticos (enviados por el sistema) o manuales (enviados por otro usuario).

CU8.2 Leer aviso

El usuario podrá leer el aviso recibido.

CU8.3 Responder aviso

El usuario podrá responder el aviso recibido. Para ello deberá indicar el contenido de la respuesta.

CU8.4 Crear aviso

El usuario podrá crear un nuevo aviso, por lo que deberá indicar el destinatario del aviso y el contenido del mismo.

CU8.5 Eliminar aviso

El usuario podrá eliminar permanentemente el aviso.

3.4. Requisitos suplementarios

A continuación se listan los requisitos suplementarios del sistema. Los requisitos estarán clasificados por tipo y subtipo. Los tipos y subtipos disponibles son los siguientes:

- **Funcionalidad:** indica que el requisito especifica una funcionalidad del sistema. Este requisito tendrá una estrecha relación con la interacción del usuario con el sistema y por tanto deberá mantener una dependencia con uno o más casos de uso. Esta dependencia se mostrará más adelante mediante la matriz de trazabilidad. Los subtipos de los que dispone son los siguientes:
 - **Básica:** este subtipo indica que el requisito especificará funcionalidad básica, que será desarrollada en la primera iteración del [Plan de Iteraciones](#).
 - **Adicional:** indica que el requisito especificará funcionalidad adicional, que será desarrollada en la segunda iteración del Plan de Iteraciones.
 - **Gráficas y documentos:** indica que el requisito especificará funcionalidad correspondiente a la realización de gráficas y generación de documentos, que será desarrollada en la tercera iteración del Plan de Iteraciones.
 - **Extra:** indica que el requisito especificará funcionalidad extra, que será desarrollada en la cuarta y última iteración del Plan de Iteraciones.
- **Interfaces:** un requisito de este tipo especifica las conexiones de entrada/salida de los datos al/del sistema. Aunque [RUP](#) destaca varios subtipos para este tipo como “De sistema”, “Externas”, “De usuario”, “Comunicaciones”, “Hardware” o “Software”, en este proyecto únicamente se utilizará el subtipo “Externas”:
 - **Externas:** este subtipo define comunicaciones externas al sistema que proveen al mismo de datos de entrada para completar la funcionalidad total de la futura aplicación. De igual forma, el sistema en desarrollo proveerá datos de salida dirigidos a dicha interfaz externa.
- **Restricciones de diseño:** los requisitos de este tipo especifican limitaciones de diseño que el sistema debe cumplir en el momento de su construcción.
- **Seguridad:** los requisitos de seguridad indican controles o medidas de seguridad que debe implantar o respetar el sistema para asegurar la integridad, disponibilidad y confidencialidad de los datos tratados y almacenados.

De forma adicional, por cada requisito se indicará el nivel de impacto y el nivel de esfuerzo. El impacto indica la importancia que tiene el requisito a la hora de su implantación al sistema actual. El esfuerzo indica la complejidad para cumplir dicho objetivo. Los niveles de medida serán: “Muy bajo”, “Bajo”, “Medio”, “Alto” y “Muy alto”.

Por otra parte, se indicará la versión final de cada requisito. Cada vez que el requisito ha sido editado, su versión ha aumentado en una unidad.

Los requisitos que deberá cumplir el sistema en desarrollo son los siguientes:

R1: Gestionar proyectos			
Descripción:	El sistema debe permitir al usuario crear, editar y eliminar proyectos.		
Tipo:	Funcionalidad	Subtipo:	Básica
Impacto:	Muy alto	Esfuerzo:	Medio
Versión final:	2		

Tabla 8. R1: Gestionar proyectos.

R2: Gestionar iteraciones			
Descripción:	El sistema debe permitir al usuario crear, editar y eliminar iteraciones clasificadas por fases.		
Tipo:	Funcionalidad	Subtipo:	Básica
Impacto:	Alto	Esfuerzo:	Medio
Versión final:	2		

Tabla 9. R2: Gestionar iteraciones.

R3: Gestionar requisitos			
Descripción:	El sistema debe permitir al usuario crear, editar y eliminar requisitos.		
Tipo:	Funcionalidad	Subtipo:	Básica
Impacto:	Muy alto	Esfuerzo:	Medio
Versión final:	1		

Tabla 10. R3: Gestionar requisitos.

R4: Gestionar casos de uso			
Descripción:	El sistema debe permitir al usuario crear, editar y eliminar casos de uso.		
Tipo:	Funcionalidad	Subtipo:	Básica
Impacto:	Muy alto	Esfuerzo:	Medio
Versión final:	1		

Tabla 11. R4: Gestionar casos de uso.

R5: Gestionar trazabilidad			
Descripción:	El sistema permitirá al usuario establecer relaciones de dependencia entre requisitos y casos de uso a través de una matriz de trazabilidad.		
Tipo:	Funcionalidad	Subtipo:	Básica
Impacto:	Alto	Esfuerzo:	Alto
Versión final:	1		

Tabla 12. R5: Gestionar trazabilidad.

R6: Gestionar entregables			
Descripción:	El sistema debe permitir al usuario crear, editar y eliminar entregables.		
Tipo:	Funcionalidad	Subtipo:	Básica
Impacto:	Muy alto	Esfuerzo:	Alto
Versión final:	1		

Tabla 13. R6: Gestionar entregables.

R7: Datos casos de uso			
Descripción:	El sistema obligará al usuario a indicar un título y una imagen asociada al caso de uso que se creará o editará. En caso de no indicar alguno de estos campos, el sistema avisará al usuario a través de una alerta visual.		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Medio	Esfuerzo:	Medio
Versión final:	2		

Tabla 14. R7: Datos casos de uso.

R8: Generación de entregables			
Descripción:	El sistema permitirá generar de dos formas los entregables: de forma manual y de forma automática.		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Medio	Esfuerzo:	Alto
Versión final:	2		

Tabla 15. R8: Generación de entregables.

R9: Generación manual de entregables			
Descripción:	El sistema permitirá que el usuario adjunte archivos externos a la aplicación para generar un nuevo entregable. Éste consistirá en la consecución de los archivos indicados anteriormente.		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Medio	Esfuerzo:	Muy alto
Versión final:	2		

Tabla 16. R9: Generación manual de entregables.

R10: Generación automática de entregables			
Descripción:	El sistema permitirá al usuario obtener un documento generado de forma automática a partir de los datos internos del proyecto en desarrollo. De igual forma, permitirá incluir documentos adjuntos. Estos documentos se presentarán a continuación del contenido generado de forma automática.		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Medio	Esfuerzo:	Muy alto
Versión final:	2		

Tabla 17. R10: Generación automática de entregables.

R11: Formato de documentos			
Descripción:	El formato de documento que el usuario podrá incluir para la generación de entregables será uno de los siguientes: .txt, .doc, .docx o .pdf.		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Medio	Esfuerzo:	Medio
Versión final:	3		

Tabla 18. R11: Formato de documentos.

R12: Formato de entregables			
Descripción:	El formato que tendrán los entregables generados por el sistema será .doc.		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Medio	Esfuerzo:	Medio
Versión final:	2		

Tabla 19. R12: Formato de entregables.

R13: Datos de iteraciones			
Descripción:	El sistema obligará al usuario a indicar un nombre y una duración (en semanas) asociada a la iteración que se creará o editará. En caso de no indicar alguno de estos campos, el sistema avisará al usuario a través de una alerta visual. De forma opcional, el sistema permitirá al usuario seleccionar o deseleccionar los productos desde un listado predefinido. Por cada producto, el usuario podrá asignar o desasignar un responsable.		
Tipo:	Funcionalidad	Subtipo:	Básica
Impacto:	Alto	Esfuerzo:	Muy alto
Versión final:	2		

Tabla 20. R13: Datos de iteraciones.

R14: Datos de proyectos			
Descripción:	El sistema obligará al usuario a indicar un nombre y una descripción asociados al proyecto que se creará o editará. En caso de no indicar alguno de estos campos, el sistema avisará al usuario a través de una alerta visual.		
Tipo:	Funcionalidad	Subtipo:	Básica
Impacto:	Medio	Esfuerzo:	Medio
Versión final:	1		

Tabla 21. R14: Datos de proyectos.

R15: Datos de requisitos			
Descripción:	A la hora de crear o editar un requisito, el sistema obligará al usuario a indicar un tipo de trazabilidad, un título, una descripción y un nivel de impacto y esfuerzo. En caso de no indicar alguno de estos campos, el sistema avisará al usuario a través de una alerta visual. De forma opcional, el usuario podrá indicar el tipo y subtipo de requisito que está gestionado.		
Tipo:	Funcionalidad	Subtipo:	Básica
Impacto:	Medio	Esfuerzo:	Alto
Versión final:	1		

Tabla 22. R15: Datos de requisitos.

R16: Tipos de requisitos			
Descripción:	El usuario podrá seleccionar el tipo de requisito desde una lista predefinida. El sistema dispondrá de una opción adicional, que permitirá al usuario crear un nuevo tipo de requisito. Dicho tipo se guardará de tal forma que el usuario pueda seleccionarlo desde la lista de tipos en gestiones futuras.		
Tipo:	Funcionalidad	Subtipo:	Básica
Impacto:	Alto	Esfuerzo:	Muy alto
Versión final:	1		

Tabla 23. R16: Tipos de requisitos.

R17: Subtipos de requisitos			
Descripción:	El usuario podrá seleccionar el subtipo de requisito desde una lista predefinida correspondiente al tipo seleccionado. El sistema dispondrá de una opción adicional, que permitirá al usuario crear un nuevo subtipo de requisito. Dicho subtipo se guardará de tal forma que el usuario pueda seleccionarlo desde la lista de subtipos (correspondiente al tipo seleccionado) en gestiones futuras.		
Tipo:	Funcionalidad	Subtipo:	Básica
Impacto:	Medio	Esfuerzo:	Muy alto
Versión final:	1		

Tabla 24. R17: Subtipos de requisitos.

R18: Interfaz de usuarios			
Descripción:	El sistema gestionará a los usuarios que pertenezcan a la interfaz pública de usuarios que se ha proporcionado.		
Tipo:	Interfases	Subtipo:	Externas
Impacto:	Medio	Esfuerzo:	Medio
Versión final:	2		

Tabla 25. R18: Interfaz de usuarios.

R19: Clasificación de usuarios			
Descripción:	El sistema permitirá una serie de acciones dependiendo del rol de los usuarios indicados en la interfaz de usuarios externa. (<i>Ver requisitos de R63 a R67</i>).		
Tipo:	Interfases	Subtipo:	Externas
Impacto:	Alto	Esfuerzo:	Alto
Versión final:	1		

Tabla 26. R19: Clasificación de usuarios.

R20: Gestionar participantes en un proyecto			
Descripción:	El sistema permitirá al usuario asignar a otros usuarios como participantes de un proyecto. El usuario deberá asignar a cada participante añadido un rol específico para el proyecto.		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Medio	Esfuerzo:	Alto
Versión final:	1		

Tabla 27. R20: Gestionar participantes en un proyecto.

R21: Listado de proyectos			
Descripción:	El sistema mostrará al usuario una tabla con el listado de proyectos a los que pertenezca. Por cada proyecto se mostrará el título, la descripción y las acciones que pueda realizar sobre él (edición, historial, descarga, asignación de usuarios y/o borrado).		
Tipo:	Funcionalidad	Subtipo:	Básica
Impacto:	Bajo	Esfuerzo:	Bajo
Versión final:	2		

Tabla 28. R21: Listado de proyectos.

R22: Listado de iteraciones			
Descripción:	El sistema mostrará al usuario una tabla con el listado de iteraciones del proyecto seleccionado. Por cada iteración se mostrará el identificador, el título, los productos y las acciones que pueda realizar sobre la misma (edición, historial, y/o borrado).		
Tipo:	Funcionalidad	Subtipo:	Básica
Impacto:	Bajo	Esfuerzo:	Bajo
Versión final:	1		

Tabla 29. R22: Listado de iteraciones.

R23: Listado de requisitos			
Descripción:	El sistema mostrará al usuario una tabla con el listado de requisitos del proyecto seleccionado. Por cada requisito se mostrará el identificador, el título, la descripción y otros detalles del requisito. Asimismo, el usuario tendrá disponible las acciones que pueda realizar sobre él (edición, historial, trazabilidad, publicación, aceptación, rechazo y/o borrado).		
Tipo:	Funcionalidad	Subtipo:	Básica
Impacto:	Bajo	Esfuerzo:	Bajo
Versión final:	1		

Tabla 30. R23: Listado de requisitos.

R24: Listado de casos de uso			
Descripción:	El sistema mostrará al usuario una tabla con el listado de casos de uso del proyecto seleccionado. Por cada caso de uso se mostrará el identificador y otros detalles del mismo. Asimismo, el usuario tendrá disponible las acciones que pueda realizar sobre caso de uso (edición, historial, publicación, aceptación, rechazo y/o borrado).		
Tipo:	Funcionalidad	Subtipo:	Básica
Impacto:	Bajo	Esfuerzo:	Bajo
Versión final:	1		

Tabla 31. R24: Listado de casos de uso.

R25: Listado de entregables			
Descripción:	El usuario podrá observar el listado con todos los entregables creados. El listado de los entregables mostrará por cada fila toda la información de cada entregable creado. Además dispondrá de una serie de acciones para realizar sobre el entregable seleccionado (actualizar, publicar, descargar, eliminar, aceptar y/o rechazar).		
Tipo:	Funcionalidad	Subtipo:	Básica
Impacto:	Bajo	Esfuerzo:	Bajo
Versión final:	1		

Tabla 32. R25: Listado de entregables.

R26: Listado de avisos			
Descripción:	El sistema permitirá que el usuario observe todos sus avisos en una tabla. Por cada aviso, se mostrará el emisor, el asunto, el mensaje y otros detalles. Asimismo, el usuario tendrá disponible varias acciones (responder y eliminar).		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Bajo	Esfuerzo:	Bajo
Versión final:	1		

Tabla 33. R26: Listado de avisos.

R27: Publicar requisito			
Descripción:	El sistema pondrá a disposición del usuario un botón para publicar un requisito. Dicho requisito pasará al estado "Publicado" (<i>Ver requisito R30</i>).		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Medio	Esfuerzo:	Medio
Versión final:	2		

Tabla 34. R27: Publicar requisito.

R28: Publicar caso de uso			
Descripción:	El sistema pondrá a disposición del usuario un botón para publicar un caso de uso. Dicho caso de uso pasará al estado "Publicado" (<i>Ver requisito R30</i>).		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Medio	Esfuerzo:	Medio
Versión final:	2		

Tabla 35. R28: Publicar caso de uso.

R29: Publicar entregable			
Descripción:	El sistema pondrá a disposición del usuario un botón para publicar un entregable. Dicho entregable pasará al estado "Publicado" (<i>Ver requisito R30</i>).		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Medio	Esfuerzo:	Medio
Versión final:	1		

Tabla 36. R29: Publicar entregable.

R30: Estados de elementos			
Descripción:	Cada requisito, caso de uso y entregable dispondrá de un ciclo de vida representado por varios estados: Procesado, revisando, publicado, rechazado y aceptado.		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Alto	Esfuerzo:	Medio
Versión final:	1		

Tabla 37. R30: Estados de elementos.

R31: Estado procesado			
Descripción:	Los requisitos, casos de usos o entregables podrán tener el estado "Procesado". Este estado indica que el elemento ha sido creado y está a la espera de ser publicado o editado. Desde este estado se permitirá realizar cualquier acción sobre el elemento (<i>Ver requisitos de R63 a R67</i>). El estado estará representado por un icono de reloj de arena.		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Medio	Esfuerzo:	Medio
Versión final:	1		

Tabla 38. R31: Estado procesado.

R32: Estado revisando			
Descripción:	Los requisitos, casos de usos o entregables podrán tener el estado "Revisando". Los elementos pasarán a este estado cuando sean enviados a evaluación al Project Manager. Este estado perdurará hasta que el Project Manager publique el elemento al cliente. De igual forma, si un usuario edita elemento, el sistema avisará con una alerta, ya que el estado volverá a "Procesado" tras su edición. El estado "Revisando" estará representado por un icono de interrogación. El Project Manager recibirá un aviso de nuevo requisito, caso de uso o entregable listo para evaluar.		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Medio	Esfuerzo:	Medio
Versión final:	1		

Tabla 39. R32: Estado revisando.

R33: Estado publicado			
Descripción:	Los requisitos, casos de usos o entregables podrán tener el estado "Publicado". Este estado indicará que el elemento ha sido publicado por el Project Manager al cliente. Este requisito, caso de uso o entregable pasará a estar visible en la zona del cliente para que éste pueda aceptarlo o rechazarlo. De igual forma, el cliente recibirá un aviso con la nueva publicación. Este estado perdurará hasta que el cliente acepte o rechace el elemento o bien un usuario lo edite. El estado "Publicado" estará representado por un icono de un hombre con traje.		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Alto	Esfuerzo:	Alto
Versión final:	1		

Tabla 40. R33: Estado publicado.

R34: Estado rechazado			
Descripción:	Los requisitos, casos de usos o entregables podrán tener el estado "Rechazado". Este estado indicará que el propietario ha rechazado el requisito, caso de uso o entregable que fue publicado. Al realizar esta opción, el sistema proporcionará al cliente un campo de texto para que indique las razones de su rechazo. Estas razones se incluirán en el aviso que llegará al Project Manager. Este estado perdurará hasta que el requisito o caso de uso sea editado, para volver a comenzar su ciclo de vida desde el estado "Procesado". El estado "Rechazado" estará representado por un icono de una cruz roja.		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Alto	Esfuerzo:	Muy alto
Versión final:	1		

Tabla 41. R34: Estado rechazado.

R35: Estado aceptado			
Descripción:	Los requisitos, casos de usos o entregables podrán tener el estado "Aceptado". Este estado indicará que el propietario ha aceptado el requisito, caso de uso o entregable. Al aceptar el elemento, el sistema avisará al Project Manager con la decisión tomada. El estado "Aceptado" es el último en el ciclo de vida de un elemento. Si se trata de un requisito o caso de uso, podrá ser añadido a los entregables finales correspondientes. El estado "Aceptado" está representado por un icono de "tick" verde.		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Alto	Esfuerzo:	Muy alto
Versión final:	1		

Tabla 42. R35: Estado aceptado.

R36: Aceptar requisito			
Descripción:	El usuario dispondrá de un botón para aceptar un requisito que ha sido publicado. Tras pulsar el botón, el requisito pasará al estado "Aceptado" (<i>Ver requisito R30</i>).		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Medio	Esfuerzo:	Alto
Versión final:	1		

Tabla 43. R36: Aceptar requisito.

R37: Aceptar caso de uso			
Descripción:	El usuario dispondrá de un botón para aceptar un caso de uso que ha sido publicado. Tras pulsar el botón, el caso de uso pasará al estado "Aceptado" (<i>Ver requisito R30</i>).		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Medio	Esfuerzo:	Alto
Versión final:	1		

Tabla 44. R37: Aceptar caso de uso.

R38: Aceptar entregable			
Descripción:	El usuario dispondrá de un botón para aceptar un entregable que ha sido publicado. Tras pulsar el botón, el entregable pasará al estado "Aceptado" (<i>Ver requisito R30</i>).		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Medio	Esfuerzo:	Alto
Versión final:	2		

Tabla 45. R38: Aceptar entregable.

R39: Rechazar requisito			
Descripción:	El usuario dispondrá de un botón para rechazar un requisito que ha sido publicado. Tras pulsar el botón, el sistema mostrará un campo de texto para que el usuario, de forma opcional, indique el motivo del rechazo. Tras pulsa un botón de confirmación, el requisito pasará al estado "Rechazado" (<i>Ver requisito R30</i>).		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Medio	Esfuerzo:	Alto
Versión final:	1		

Tabla 46. R39: Rechazar requisito.

R40: Rechazar caso de uso			
Descripción:	El usuario dispondrá de un botón para rechazar un caso de uso que ha sido publicado. Tras pulsar el botón, el sistema mostrará un campo de texto para que el usuario, de forma opcional, indique el motivo del rechazo. Tras pulsa un botón de confirmación, el caso de uso pasará al estado "Rechazado" (<i>Ver requisito R30</i>).		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Medio	Esfuerzo:	Alto
Versión final:	1		

Tabla 47. R40: Rechazar caso de uso.

R41: Rechazar entregable			
Descripción:	El usuario dispondrá de un botón para rechazar un entregable que ha sido publicado. Tras pulsar el botón, el sistema mostrará un campo de texto para que el usuario, de forma opcional, indique el motivo del rechazo. Tras pulsa un botón de confirmación, el entregable pasará al estado "Rechazado" (<i>Ver requisito R30</i>).		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Medio	Esfuerzo:	Alto
Versión final:	2		

Tabla 48. R41: Rechazar entregable.

R42: Gestionar avisos			
Descripción:	El sistema permitirá al usuario crear, responder, recibir y eliminar avisos.		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Bajo	Esfuerzo:	Medio
Versión final:	1		

Tabla 49. R42: Gestionar avisos.

R43: Datos de avisos			
Descripción:	El sistema obligará al usuario a indicar un destinatario y un mensaje asociado al aviso que se creará. En caso de respuesta, el usuario sólo deberá indicar un mensaje ya que el destinatario se incluirá de forma automática. En caso de no indicar alguno de estos campos, el sistema avisará al usuario a través de una alerta visual.		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Medio	Esfuerzo:	Medio
Versión final:	1		

Tabla 50. R43: Datos de avisos.

R44: Descargar proyecto			
Descripción:	El usuario podrá descargar todos los entregables generados durante el desarrollo del proyecto desde la lista de proyectos. Los entregables se descargarán comprimidos en un fichero .zip.		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Bajo	Esfuerzo:	Bajo
Versión final:	1		

Tabla 51. R44: Descargar proyecto.

R45: Historial del proyecto			
Descripción:	El usuario tendrá un control absoluto sobre las acciones que se realicen en el proyecto. Toda acción quedará registrada en el historial, indicando quién lo ha realizado, el qué ha realizado, qué valor ha sufrido la acción y la fecha en la que se realizó.		
Tipo:	Funcionalidad	Subtipo:	Básica
Impacto:	Bajo	Esfuerzo:	Medio
Versión final:	1		

Tabla 52. R45: Historial del proyecto.

R46: Gráficas de un proyecto			
Descripción:	El sistema permitirá a los usuarios percibir de forma gráfica un resumen del estado actual del proyecto. Para ello tendrá disponible un listado de gráficas que podrá añadir o eliminar (<i>Ver requisitos de R47 a 58</i>).		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Medio	Esfuerzo:	Muy alto
Versión final:	1		

Tabla 53. R46: Gráficas de un proyecto.

R47: Gráfica "Número de requisitos suplementarios por estado"			
Descripción:	El usuario dispondrá de una gráfica del tipo circular en 3D que mostrará los porcentajes del número de requisitos creados en un proyecto, clasificados por los diferentes estados de su ciclo de vida.		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Alto	Esfuerzo:	Medio
Versión final:	1		

Tabla 54. R47: Gráfica "Número de requisitos suplementarios por estado".

R48: Gráfica "Número de iteraciones por estado y fases"			
Descripción:	El usuario podrá observar una gráfica de barras en el que se indicará el número de iteraciones (eje Y) por cada fase (eje X). Además cada barra dispondrá de un color: azul si la iteración tiene estado "Pendiente" o roja si su estado es "Cerrada".		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Alto	Esfuerzo:	Medio
Versión final:	1		

Tabla 55. R48: Gráfica "Número de iteraciones por estado y fases".

R49: Gráfica "Requisitos editados por semana"			
Descripción:	El usuario podrá observar una gráfica de barras que indicará el número de ediciones sobre requisitos (eje Y) que se han realizado por semana (eje X).		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Alto	Esfuerzo:	Medio
Versión final:	1		

Tabla 56. R49: Gráfica "Requisitos editados por semana".

R50: Gráfica "Estabilidad media de requisitos por semana"			
Descripción:	El usuario podrá observar una gráfica de barras que indicará por cada semana (eje X) la estabilidad media conjunta de todos los requisitos (eje Y). La estabilidad media será calculada realizando la media aritmética de la estabilidad de cada requisito. La estabilidad de cada requisito se calculará por cada edición que sufra de la siguiente forma: $\text{Estabilidad} = 100 - (\text{N_ediciones_req} / \text{N_ediciones_totales})$, donde N_ediciones_req es el número de ediciones del requisito editado y N_ediciones_totales es el número de ediciones de todos los requisitos. Así, un requisito recién creado será 100% estable, pero conforme se edita y se generan nuevas versiones del mismo, la estabilidad irá disminuyendo teniendo en cuenta las ediciones del resto de requisitos.		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Alto	Esfuerzo:	Alto
Versión final:	1		

Tabla 57. R50: Gráfica "Estabilidad media de requisitos por semana".

R51: Gráfica "Número de iteraciones por estado"			
Descripción:	El usuario podrá observar una gráfica del tipo circular en 3D que mostrará los porcentajes del número de iteraciones creadas por los diferentes estados.		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Medio	Esfuerzo:	Medio
Versión final:	1		

Tabla 58. R51: Gráfica "Número de iteraciones por estado".

R52: Gráfica "Casos de uso por estado"			
Descripción:	El usuario podrá observar una gráfica del tipo circular en 3D que muestra los porcentajes del número de casos de uso por los diferentes estados.		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Medio	Esfuerzo:	Medio
Versión final:	1		

Tabla 59. R52: Gráfica "Casos de uso por estado".

R53: Gráfica "Entregables por estado"			
Descripción:	El usuario podrá observar una gráfica del tipo circular en 3D que mostrará los porcentajes del número de entregables por los diferentes estados.		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Medio	Esfuerzo:	Medio
Versión final:	4		

Tabla 60. R53: Gráfica "Entregables por estado".

R54: Gráfica "Casos de uso editados por semana"			
Descripción:	El usuario podrá observar una gráfica de barras que indicará el número de ediciones sobre casos de uso (eje Y) que se han realizado por semana (eje X).		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Medio	Esfuerzo:	Alto
Versión final:	1		

Tabla 61. R54: Gráfica "Casos de uso editados por semana".

R55: Gráfica "Entregables editados por semana"			
Descripción:	El usuario podrá observar una gráfica de barras que indicará el número de actualizaciones sobre entregables (eje Y) que se han realizado por semana (eje X).		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Medio	Esfuerzo:	Alto
Versión final:	1		

Tabla 62. R55: Gráfica "Entregables editados por semana".

R56: Gráfica "Requisitos por versión"			
Descripción:	El usuario podrá observar una gráfica del tipo circular en 3D que mostrará los porcentajes del número de requisitos por los diferentes límites de versiones (1, de 2 a 5, de 6 a 10, de 11 a 50 y más de 50).		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Medio	Esfuerzo:	Alto
Versión final:	1		

Tabla 63. R56: Gráfica "Requisitos por versión".

R57: Gráfica "Casos de uso por versión"			
Descripción:	El usuario podrá observar una gráfica del tipo circular en 3D que mostrará los porcentajes del número de casos de uso por los diferentes límites de versiones (1, de 2 a 5, de 6 a 10, de 11 a 50 y más de 50).		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Medio	Esfuerzo:	Alto
Versión final:	1		

Tabla 64. R57: Gráfica "Casos de uso por versión".

R58: Gráfica "Entregables por versión"			
Descripción:	El usuario podrá observar una gráfica del tipo circular en 3D que mostrará los porcentajes del número de entregables por los diferentes límites de versiones (1, de 2 a 5, de 6 a 10, de 11 a 50 y más de 50).		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Medio	Esfuerzo:	Alto
Versión final:	1		

Tabla 65. R58: Gráfica "Entregables por versión".

R59: Gestión de gráficas			
Descripción:	El usuario podrá acceder a un panel de personalización que le permitirá activar y desactivar las gráficas que visualizará desde la pantalla principal de un proyecto en concreto.		
Tipo:	Funcionalidad	Subtipo:	Gráficas y documentos
Impacto:	Bajo	Esfuerzo:	Bajo
Versión final:	1		

Tabla 66. R59: Gestión de gráficas.

R60: Ayuda al usuario			
Descripción:	El usuario podrá acceder a una sección de ayuda tipo "Wiki" (referencia) desde la que accederá a explicaciones sobre todas las acciones que pueda realizar en la aplicación.		
Tipo:	Funcionalidad	Subtipo:	Extra
Impacto:	Muy bajo	Esfuerzo:	Medio
Versión final:	1		

Tabla 67. R60: Ayuda al usuario.

R61: Historial del requisito			
Descripción:	El usuario podrá ver todas las acciones que se han realizado sobre el requisito seleccionado. Por cada acción el sistema mostrará quién la llevó a cabo y cuándo fue realizada.		
Tipo:	Funcionalidad	Subtipo:	Básica
Impacto:	Bajo	Esfuerzo:	Medio
Versión final:	1		

Tabla 68. R61: Historial del requisito.

R62: Historial del caso de uso			
Descripción:	El usuario podrá ver todas las acciones que se han realizado sobre el caso de uso seleccionado. Por cada acción el sistema mostrará quién la llevó a cabo y cuándo fue realizada.		
Tipo:	Funcionalidad	Subtipo:	Básica
Impacto:	Bajo	Esfuerzo:	Medio
Versión final:	1		

Tabla 69. R62: Historial del caso de uso.

R63: Permiso de Administrador			
Descripción:	El usuario con permiso "Administrador" podrá acceder a todas las áreas del programa, así como realizar cualquier acción de lectura, creación, edición o borrado de cualquier elemento.		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Medio	Esfuerzo:	Alto
Versión final:	1		

Tabla 70. R63: Permiso de Administrador.

R64: Permiso de Propietario			
Descripción:	El usuario con permiso "Propietario" podrá acceder únicamente a documentación publicada y al historial del proyecto. Dispondrá de las siguientes acciones permitidas: aceptación o rechazo de requisitos, casos de uso y entregables.		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Alto	Esfuerzo:	Alto
Versión final:	1		

Tabla 71. R64: Permiso de Propietario.

R65: Permiso de Project Manager			
Descripción:	El usuario podrá acceder a todas las áreas dentro del proyecto al que pertenezca. Acciones permitidas: Asignación de usuarios al proyecto, asignación de roles a los usuarios participantes, creación y edición del Plan de Iteraciones, lectura, creación, edición y publicación de requisitos, casos de uso y entregables, descarga de entregables y del proyecto completo y edición de las propiedades del proyecto.		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Alto	Esfuerzo:	Alto
Versión final:	1		

Tabla 72. R65: Permiso de Project Manager.

R66: Permiso de Ingeniero de Requisitos			
Descripción:	El usuario tendrá acceso a todas las áreas del proyecto en modo lectura, excepto "Gestión de usuarios" y "Edición de un proyecto". Acciones permitidas: Lectura del Plan de Iteraciones, lectura, creación, edición y publicación al Project Manager de requisitos y casos de uso y lectura de entregables.		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Alto	Esfuerzo:	Alto
Versión final:	1		

Tabla 73. R66: Permiso de Ingeniero de Requisitos.

R67: Permiso de Gestor de entregables			
Descripción:	El usuario tendrá acceso a todas las áreas del proyecto en modo lectura, excepto "Gestión de usuarios" y "Edición del proyecto". Acciones permitidas: Lectura del Plan de Iteraciones, lectura, creación, edición y publicación al Project Manager de entregables y lectura de requisitos y casos de uso.		
Tipo:	Funcionalidad	Subtipo:	Adicional
Impacto:	Alto	Esfuerzo:	Alto
Versión final:	2		

Tabla 74. R67: Permiso de Gestor de entregables.

R68: Seguridad frente ataques			
Descripción:	El sistema deberá proteger los datos sensibles a ataques como Brute Force o SQL Injection .		
Tipo:	Seguridad	Subtipo:	-
Impacto:	Medio	Esfuerzo:	Alto
Versión final:	1		

Tabla 75. R68: Seguridad frente ataques.

R69: Control de acceso			
Descripción:	El sistema deberá controlar el acceso de los usuarios, identificando a cada usuario permitido junto con roles que dispone por proyecto.		
Tipo:	Seguridad	Subtipo:	-
Impacto:	Medio	Esfuerzo:	Medio
Versión final:	1		

Tabla 76. R69: Control de acceso.

R70: Compatibilidad con navegadores web			
Descripción:	La aplicación será compatible con los siguientes navegadores de internet: Mozilla Firefox, versión 3.6 en adelante; Internet Explorer, versión 8.0 en adelante; Google Chrome, versión 7.0 en adelante.		
Tipo:	Restricciones de diseño	Subtipo:	-
Impacto:	Medio	Esfuerzo:	Alto
Versión final:	1		

Tabla 77. R70: Compatibilidad con navegadores web.

R71: Garantizar el cumplimiento de la Ley Orgánica de Protección de Datos (LOPD)			
Descripción:	La aplicación deberá contar con las medidas de seguridad necesarias para asegurar el cumplimiento de la Ley Orgánica de Protección de Datos en lo relativo a los datos personales de los usuarios almacenados en el sistema.		
Tipo:	Seguridad	Subtipo:	-
Impacto:	Medio	Esfuerzo:	Medio
Versión final:	1		

Tabla 78. R71: Garantizar el cumplimiento de la Ley Orgánica de Protección de Datos ([LOPD](#)).

R72: Diseño de la aplicación			
Descripción:	El diseño de la aplicación será lo más sencillo posible. Se mantendrá una gama de colores azules y grises para los botones, textos y demás objetos.		
Tipo:	Restricciones de diseño	Subtipo:	-
Impacto:	Medio	Esfuerzo:	Medio
Versión final:	1		

Tabla 79. R72: Diseño de la aplicación.

R73: Menú principal			
Descripción:	Lo primero que el usuario observará tras identificarse en la aplicación será el menú principal con las opciones "Gestión de proyectos", "Personalización" y "Ayuda".		
Tipo:	Restricciones de diseño	Subtipo:	-
Impacto:	Medio	Esfuerzo:	Muy bajo
Versión final:	1		

Tabla 80. R73: Menú principal.

R74: Estructura de las secciones			
Descripción:	Todas las secciones de la aplicación dispondrán del mismo esqueleto. Éste consistirá en una barra superior que contará con los siguientes elementos en el orden en que se mencionan: Enlace al menú principal (texto "Home"), selector de proyectos en los que participa el usuario, selector de avisos o salida y la hora actual. La excepción es la pantalla de login, que contará únicamente con los elementos "Home" y la hora actual.		
Tipo:	Restricciones de diseño	Subtipo:	-
Impacto:	Medio	Esfuerzo:	Medio
Versión final:	3		

Tabla 81. R74: Estructura de las secciones.

R75: Selector de proyectos			
Descripción:	El selector de proyectos consistirá en un desplegable mostrando los proyectos a los que pertenece el usuario así como el rol que desempeña en cada uno. Si el usuario selecciona uno de los proyectos, se le mostrará la pantalla principal del mismo.		
Tipo:	Restricciones de diseño	Subtipo:	-
Impacto:	Medio	Esfuerzo:	Alto
Versión final:	2		

Tabla 82. R75: Selector de proyectos.

R76: Selector de avisos			
Descripción:	El selector de avisos consistirá en un desplegable situado a la derecha de la barra superior, mostrando el nombre completo del usuario. Si el usuario hace clic sobre el selector, podrá escoger las opciones "Avisos" (gestión de avisos) y "Salir" (salir del sistema).		
Tipo:	Restricciones de diseño	Subtipo:	-
Impacto:	Medio	Esfuerzo:	Medio
Versión final:	1		

Tabla 83. R76: Selector de avisos.

R77: Estructura de tablas			
Descripción:	Todas las tablas que muestren los detalles de los elementos principales de la aplicación (Requisitos, entregables, proyectos, etc.) deberán tener la misma estructura. De igual forma, los registros de las tablas se podrán ordenar de forma ascendiente o descendente por cada columna. El usuario tendrá disponible iconos que indiquen la acción bajo los títulos de cada columna.		
Tipo:	Restricciones de diseño	Subtipo:	-
Impacto:	Medio	Esfuerzo:	Alto
Versión final:	1		

Tabla 84. R77: Estructura de tablas.

R78: Estructura de la pantalla principal de un proyecto			
Descripción:	El usuario podrá observar en primer lugar las gráficas que tenga activadas. A continuación tendrá disponibles cuatro botones: "Plan de iteraciones", "Requisitos", "Trazabilidad" y "Entregables". Debajo de los botones podrá observar la fase y la iteración en las que se encuentra el proyecto, ofreciendo la posibilidad de aumentar de fase o de iteración. Finalmente, en la parte inferior de la pantalla, el usuario dispondrá de los enlaces "Usuarios" (gestión de participantes), "Propiedades" (edición del proyecto) e "Historial".		
Tipo:	Restricciones de diseño	Subtipo:	-
Impacto:	Bajo	Esfuerzo:	Bajo
Versión final:	3		

Tabla 85. R78: Estructura de la pantalla principal de un proyecto.

R79: Optimización para navegador Chrome			
Descripción:	La aplicación deberá estar perfectamente diseñada y desarrollada para que funcione de forma óptima bajo el navegador Google Chrome versión 17.		
Tipo:	Restricciones de diseño	Subtipo:	-
Impacto:	Alto	Esfuerzo:	Muy alto
Versión final:	1		

Tabla 86. R79: Optimización para navegador Chrome.

R80: Logotipo de la aplicación			
Descripción:	El logotipo de la aplicación aparecerá en la parte inferior derecha de todas las secciones excepto la pantalla de login. En este caso el logotipo aparecerá en la parte central de la pantalla. El logotipo estará inspirado en un entorno distribuido, en el que varias entidades se distribuyen circularmente en torno a un punto central.		
Tipo:	Restricciones de diseño	Subtipo:	-
Impacto:	Bajo	Esfuerzo:	Bajo
Versión final:	1		

Tabla 87. R80: Logotipo de la aplicación.

R81: Hardware de usuario			
Descripción:	El usuario deberá disponer de conexión a internet para acceder al sistema.		
Tipo:	Restricciones de diseño	Subtipo:	-
Impacto:	Bajo	Esfuerzo:	Bajo
Versión final:	1		

Tabla 88. R81: Hardware de usuario.

R82: Restricciones de usuario			
Descripción:	El usuario deberá disponer de un navegador de internet que soporte HTML4, Javascript y CSS3.		
Tipo:	Restricciones de diseño	Subtipo:	-
Impacto:	Bajo	Esfuerzo:	Bajo
Versión final:	1		

Tabla 89. R82: Restricciones de usuario.

R83: Restricciones del servidor			
Descripción:	<p>El servidor que ofrezca la aplicación deberá cumplir con unos requerimientos mínimos:</p> <ul style="list-style-type: none"> • Tener instalado un sistema operativo Linux o Windows, cuya distribución esté orientada a servir aplicaciones. • Servidor web Apache con módulo PHP 5.0 (como mínimo) instalado. • Servicio MySQL 5.0 (como mínimo) instalado. • Almacenamiento recomendado: 2TB. • Ancho de banda recomendado: 100Mbps por segundo. • Memoria RAM recomendada: 16Gb. • Disponer de acceso FTP para la carga de los ficheros de la aplicación. 		
Tipo:	Restricciones de diseño	Subtipo:	-
Impacto:	Bajo	Esfuerzo:	Medio
Versión final:	1		

Tabla 90. R83: Restricciones del servidor.

3.5. Matriz de trazabilidad

La matriz de trazabilidad permite verificar qué casos de uso quedan cubiertos por los requisitos funcionales. La matriz de trazabilidad del sistema es la siguiente:

	CU8: Gestionar avisos	CU7: Personalizar gráficas	CU6: Gestionar entregables	CU5: Gestionar trazabilidad	CU4: Gestionar casos de uso	CU3: Gestionar requisitos	CU2: Gestionar iteraciones	CU1: gestionar proyectos
R1: Gestionar proyectos								✓
R2: Gestionar iteraciones							✓	
R3: Gestionar requisitos						✓		
R4: Gestionar casos de uso					✓			
R5: Gestionar trazabilidad				✓				
R6: Gestionar entregables			✓					
R7: Datos casos de uso					✓			
R8: Generación de entregables			✓					
R9: Generación manual de entregables			✓					
R10: Generación automática de entregables			✓					
R11: Formato de documentos			✓					
R12: Formato de entregables			✓					
R13: Datos de iteraciones							✓	
R14: Datos de proyectos								✓
R15: Datos de requisitos						✓		

R16: Tipos de requisitos						✓		
R17: Subtipos de requisitos						✓		
R20: Gestionar participantes en un proyecto								✓
R21: Listado de proyectos								✓
R22: Listado de iteraciones							✓	
R23: Listado de requisitos						✓		
R24: Listado de casos de uso					✓			
R25: Listado de entregables			✓					
R26: Listado de avisos	✓							
R27: Publicar requisito						✓		
R28: Publicar caso de uso					✓			
R29: Publicar entregable			✓					
R30: Estados de elementos			✓		✓	✓		
R31: Estado procesado			✓		✓	✓		
R32: Estado revisando			✓		✓	✓		
R33: Estado publicado			✓		✓	✓		
R34: Estado rechazado			✓		✓	✓		
R35: Estado aceptado			✓		✓	✓		
R36: Aceptar requisito						✓		
R37: Aceptar caso de uso					✓			
R38: Aceptar entregable			✓					
R39: Rechazar requisito						✓		
R40: Rechazar caso de uso					✓			
R41: Rechazar entregable			✓					

R42: Gestionar avisos	✓							
R43: Datos de avisos	✓							
R44: Descargar proyecto								✓
R45: Historial del proyecto								✓
R46: Gráficas de un proyecto		✓						✓
R47: Gráfica "Número de requisitos suplementarios por estado"		✓						✓
R48: Gráfica "Número de iteraciones por estado y fases"		✓						✓
R49: Gráfica "Requisitos editados por semana"		✓						✓
R50: Gráfica "Estabilidad media de requisitos por semana"		✓						✓
R51: Gráfica "Número de iteraciones por estado"		✓						✓
R52: Gráfica "Casos de uso por estado"		✓						✓
R53: Gráfica "Entregables por estado"		✓						✓
R54: Gráfica "Casos de uso editados por semana"		✓						✓
R55: Gráfica "Entregables editados por semana"		✓						✓
R56: Gráfica "Requisitos por versión"		✓						✓
R57: Gráfica "Casos de uso por versión"		✓						✓
R58: Gráfica "Entregables por versión"		✓						✓
R59: Gestión de gráficas		✓						✓
R60: Ayuda al usuario	✓	✓	✓	✓	✓	✓	✓	✓
R61: Historial del requisito							✓	
R62: Historial del caso de uso						✓		

Tabla 91. Matriz de trazabilidad.

3.6. Análisis de la base de datos

Para que el sistema final ofrezca la funcionalidad exigida en los [casos de uso](#) y en los [requisitos suplementarios](#), es necesario analizar los tipos de elementos que estarán presentes en el sistema así como las interacciones entre ellos.

Para ello, se presenta a continuación un diagrama Entidad-Relación, gracias al cual se podrán analizar los diferentes elementos (entidades) que estarán presentes en la plataforma, así como las interacciones (relaciones) entre los mismos. De igual forma, gracias al diagrama se podrán analizar los atributos principales de cada entidad.

El diagrama Entidad-Relación ha sido elaborado mediante la herramienta [Gliffy.com](https://www.gliffy.com/) [11] y se encuentra representado en la [Figura 16 de la siguiente página](#). El sistema cuenta con un amplio número de entidades y relaciones. A continuación se ofrece el diagrama junto con una breve descripción de las principales entidades representadas.

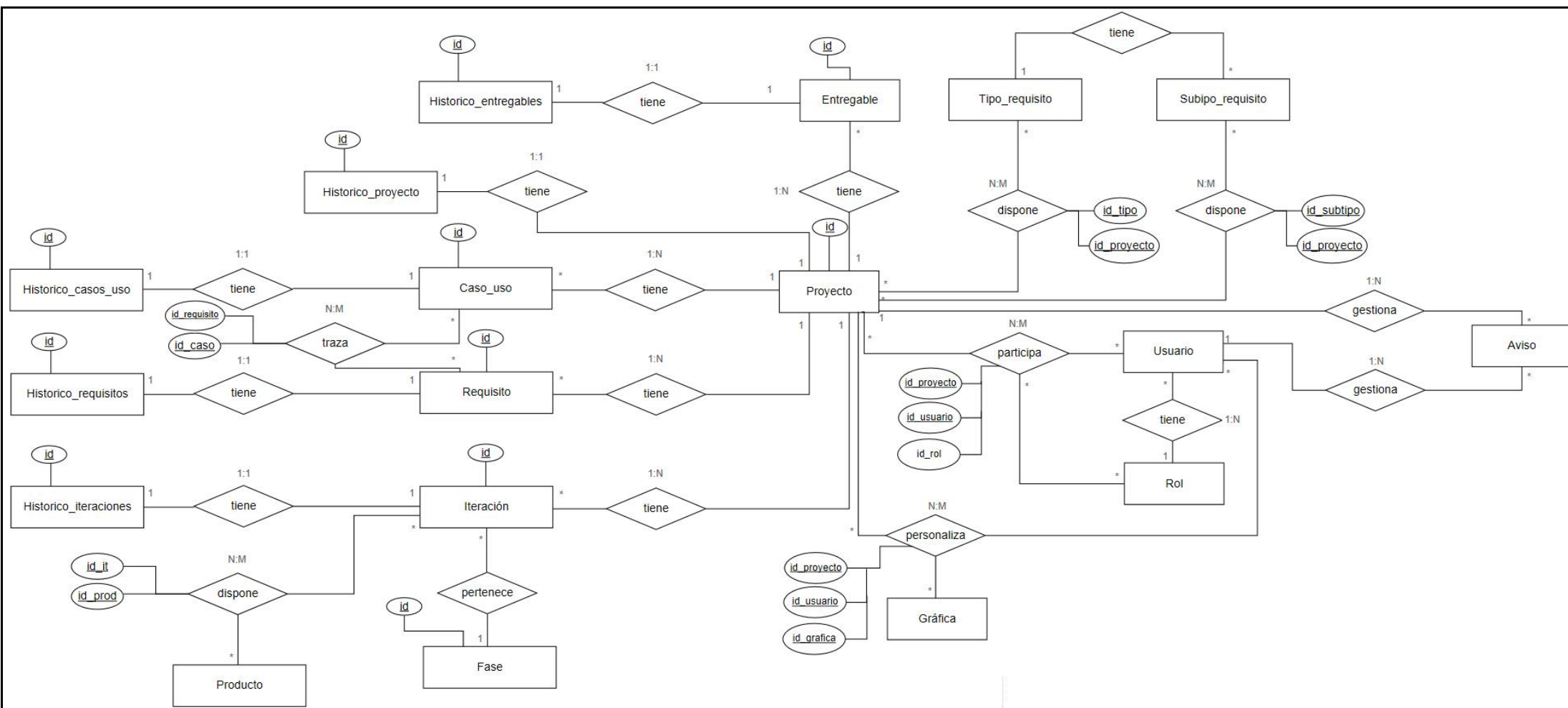


Figura 16. Diagrama Entidad Relación.

3.6.1 Entidad Proyecto

La entidad “Proyecto” es la que contiene más relaciones con el resto de entidades. En primer lugar, su atributo principal es el identificador del proyecto. En cuanto a las relaciones, son las siguientes:

- Un proyecto tiene múltiples entregables, mientras que un entregable pertenece únicamente a un proyecto. Relación “1:N”.
- Un proyecto tiene un histórico y un histórico es exclusivamente de un proyecto. Relación “1:1”.
- Un proyecto tiene varios casos de uso, mientras que un caso de uso pertenece únicamente a un proyecto concreto. Relación “1:N”.
- Un proyecto tiene muchos requisitos, mientras que un requisito pertenece a un único proyecto. Relación “1:N”.
- Un proyecto contiene múltiples iteraciones, mientras que una iteración pertenece a un solo proyecto. Relación “1:N”.
- En un proyecto participan múltiples usuarios y un usuario puede participar en varios proyectos. Relación “N:M”. La relación tiene como claves principales el identificador del proyecto y el identificador del usuario. De forma adicional, dispone del identificador de la entidad “[Rol](#)”. Esto se debe a que un usuario participa en un proyecto con un determinado rol, pudiendo ser diferente a su rol general.
- Un proyecto puede ser personalizado por múltiples usuarios y un usuario puede personalizar múltiples proyectos. Relación “N:M”. Dicha relación tiene como claves principales el identificador del proyecto, el identificador del usuario y el identificador de la entidad “[Gráfica](#)”.
- Un proyecto gestiona múltiples avisos, mientras que un aviso es gestionado únicamente por un proyecto. Relación “1:N”.
- Un proyecto dispone de varios tipos de requisitos. Dichos tipos pueden pertenecer a varios proyectos. Relación “N:M”. Las claves principales de la relación son el identificador del proyecto y el identificador del tipo de requisito.
- Un proyecto dispone de varios sub-tipos de requisitos. Dichos sub-tipos pueden pertenecer a varios proyectos. Relación “N:M”. Las claves principales de la relación son el identificador del proyecto y el identificador del sub-tipo de requisito.

3.6.2 Entidad Entregable

El atributo principal de la entidad es el identificador del entregable. En cuanto a las relaciones, son las siguientes:

- Un entregable pertenece a un proyecto, mientras que un proyecto tiene múltiples entregables. Relación “1:N”.
- Un entregable tiene un histórico. Dicho histórico pertenece a un único entregable. Relación “1:1”.

3.6.3 Entidad Caso de uso

El atributo principal de la entidad es el identificador del caso de uso. En cuanto a las relaciones, son las siguientes:

- Un caso de uso pertenece a un proyecto, mientras que un proyecto tiene múltiples casos de uso. Relación “1:N”.
- Un caso de uso tiene un histórico. Dicho histórico pertenece a un único caso de uso. Relación “1:1”.
- Un caso de uso dispone de trazabilidad con múltiples requisitos. Un requisito dispone de trazabilidad con múltiples casos de uso. Relación “N:M”. Dicha relación tiene como claves principales el identificador del caso de uso y el identificador del requisito.

3.6.4 Entidad Requisito

El atributo principal de la entidad es el identificador del requisito. En cuanto a las relaciones, son las siguientes:

- Un requisito pertenece a un proyecto, mientras que un proyecto tiene múltiples requisitos. Relación “1:N”.
- Un requisito tiene un histórico. Dicho histórico pertenece a un único requisito. Relación “1:1”.
- Un requisito dispone de trazabilidad con múltiples casos de uso. Un caso de uso dispone de trazabilidad con múltiples requisitos. Relación “N:M”. Dicha relación tiene como claves principales el identificador del caso de uso y el identificador del requisito.

3.6.5 Entidad Iteración

El atributo principal de la entidad es el identificador de la iteración. En cuanto a las relaciones, son las siguientes:

- Una iteración pertenece a un proyecto, mientras que un proyecto tiene múltiples iteraciones. Relación “1:N”.
- Una iteración tiene un histórico. Dicho histórico pertenece a una única iteración. Relación “1:1”.
- Una iteración dispone de varios tipos de productos. Un tipo de producto puede estar presente en varias iteraciones. Relación “N:M”. Dicha relación tiene como claves principales el identificador de la iteración y el identificador del tipo de producto.
- Una iteración pertenece a una fase, mientras que una fase está compuesta por múltiples iteraciones. Relación “1:N”.

3.6.6 Entidades Histórico

Todas las entidades “Histórico” cuentan con una única relación “1:1”. Esto se debe a que un histórico pertenece únicamente a un elemento y dicho elemento tiene un solo histórico.

Las entidades de este tipo son: “Histórico_proyecto”, “Histórico_entregables”, “Histórico_casos_uso”, “Histórico_requisitos” e “Histórico_iteraciones”.

3.6.7 Entidad Usuario

El atributo principal de la entidad es el identificador del usuario. En cuanto a las relaciones, son las siguientes:

- Un usuario tiene un rol, mientras que un rol puede ser obtenido por múltiples usuarios. Relación “1:N”.
- Un usuario participa en varios proyectos y en un proyecto pueden participar varios usuarios. Relación “N:M”. Dicha relación tiene como claves principales el identificador del proyecto y el identificador del usuario. De forma adicional, dispone del identificador de la entidad “[Rol](#)”. Esto se debe a que un usuario participa en un proyecto con un determinado rol, pudiendo ser diferente a su rol general.
- Un usuario gestiona múltiples avisos, mientras que un aviso pertenece exclusivamente a un usuario. Relación “1:N”.
- Un proyecto puede ser personalizado por múltiples usuarios y un usuario puede personalizar múltiples proyectos. Relación “N:M”. Dicha relación tiene como claves principales el identificador del proyecto, el identificador del usuario y el identificador de la entidad “[Gráfica](#)”.

3.6.8 Entidad Rol

El atributo principal de la entidad es el identificador del rol. En cuanto a las relaciones, son las siguientes:

- Un usuario tiene un rol, mientras que un rol pueden tenerlo múltiples usuarios. Relación “1:N”.
- Un usuario que participa en un proyecto tiene un determinado rol, mientras que dicho rol puede ser utilizado por múltiples usuarios y proyectos. Relación “N:M”. Dicha relación tiene como claves principales el identificador del proyecto y el identificador del usuario. De forma adicional, dispone del identificador de la entidad “Rol”. Esto se debe a que un usuario participa en un proyecto con un determinado rol, pudiendo ser diferente a su rol general.

3.6.9 Entidad Gráfica

El atributo principal de la entidad es el identificador de la gráfica. Esta entidad dispone de una única relación.

Una gráfica es utilizada por múltiples usuarios para personalizar múltiples proyectos. Un usuario puede personalizar un proyecto con múltiples gráficas. Relación “N:M”. Dicha relación tiene como claves principales el identificador del proyecto, el identificador del usuario y el identificador de la gráfica.

3.7. Diagrama de flujo por bloques

Tras un análisis de los casos de uso y los requisitos, se procede a la creación del diagrama del flujo de la aplicación. Dicho diagrama se adecua a las necesidades del futuro sistema en conceptos del entorno de desarrollo. Al tratarse de una aplicación [cliente-servidor](#), el diagrama se dividirá en bloques cuyas funciones son realizadas por el cliente o por el servidor.

El patrón del diagrama se corresponde con un FFBD (Functional Flow Block Diagram), creado por la Sociedad Americana de Ingenieros Mecánicos y muy utilizado en la NASA [12]. Este tipo de diagrama permite modelar y por tanto representar los diferentes procesos de un sistema especificando los módulos o bloques que generan dichos procesos.

Cada uno de los bloques tendrá un objetivo concreto y es posible que un objetivo se divida en varios subobjetivos, dando lugar a nuevos bloques, y por tanto a interacciones y funciones a desarrollar.

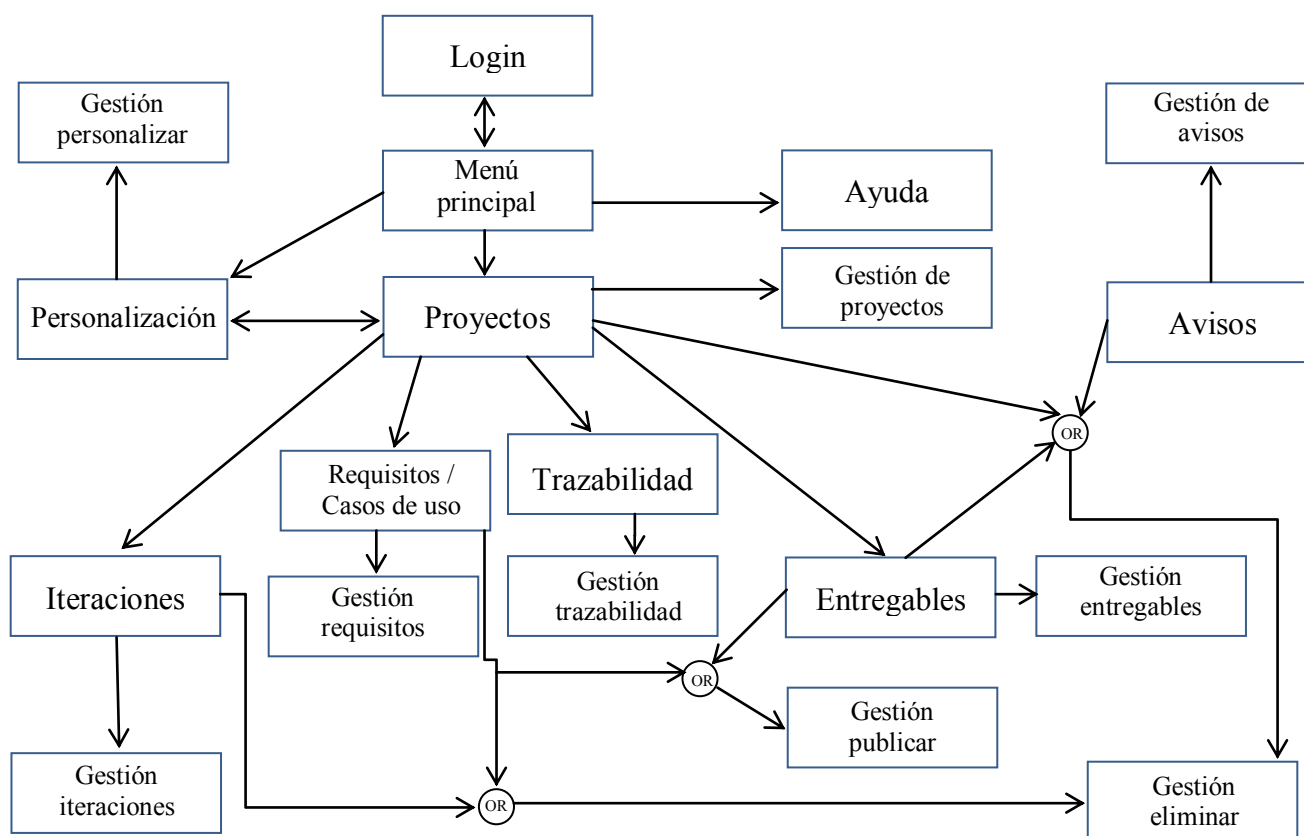


Figura 17. Diagrama de flujo del sistema.

Hay que indicar que al tratarse de una aplicación online, todos los datos se encontrarán en el servidor, por lo que es muy importante diferenciar la parte lógica de la física.

Como se observa en el diagrama, existen bloques de gestión (cuyo nombre comienza por la palabra “Gestión”) y bloques funcionales (cuyo nombre no comienza por la palabra “Gestión”). Los bloques funcionales indican, desde el punto de vista lógico, las funciones que

los usuarios tendrán disponibles al acceder a cada módulo del sistema. Dichas funciones serán realizadas de forma interna por el sistema, apoyándose en los módulos de gestión antes mencionados.

Los módulos funcionales serán los encargados de mostrar la interfaz de la aplicación al usuario, por lo que no todas las funciones irán dirigidas al mismo usuario. Éstas dependerán del rol que ejerza el usuario.

Los módulos de gestión serán los intermediarios entre el usuario y la base de datos del servidor. Serán los encargados de realizar la acción enviada por el usuario y devolver el resultado obtenido. A continuación se muestran todos los detalles de cada bloque.

3.7.1 Módulo Login

Este módulo será el encargado de gestionar los accesos a la aplicación. Las funciones que llevará a cabo son las siguientes:

- Identificar usuario.
- Obtener rol del usuario identificado.
- Obtener el listado de proyectos del usuario identificado, así como el rol específico en cada uno de ellos.

El módulo “Login” no dispone de ninguna relación de entrada, pero sí de una relación de salida. La única relación de la que dispone es con el módulo “Menú principal”. Dicho módulo será lanzado si se cumplen correctamente las funciones anteriores.

3.7.2 Módulo Menú principal

El objetivo de este módulo es proporcionar al usuario las secciones principales. Las funciones que realizará son las siguientes:

- Permitir el acceso a la gestión de proyectos.
- Permitir el acceso a la personalización de gráficas.
- Permitir el acceso a la ayuda.
- Permitir el acceso a la gestión de avisos.
- Permitir al usuario salir del sistema.

El módulo “Menú principal” dispone de una relación de entrada con el módulo “Login”. En cuanto a las relaciones de salida, el módulo dispone de una relación por cada función permitida:

- Módulo “Proyectos”.
- Módulo “Personalización”.
- Módulo “Ayuda”.

- Módulo “Avisos”.
- Módulo “Login”.

3.7.3 Módulo Proyectos

Este módulo cuenta con tres objetivos principales:

- Proporcionar al usuario el listado de los proyectos a los que pertenezca.
- Proporcionar al usuario todas las acciones posibles para gestionar un proyecto.
- Proporcionar el acceso a los datos internos de un proyecto (iteraciones, requisitos, casos de uso, trazabilidad y entregables).

Las funciones que llevará a cabo son las siguientes:

- Listar los proyectos del usuario.
- Crear nuevo proyecto.
- Editar proyecto existente.
- Eliminar proyecto existente.
- Asignar usuarios como participantes de un proyecto.
- Ver historial de un proyecto.
- Descargar los productos generados de un proyecto.

Este módulo dispone de tres relaciones de entrada:

- Módulo “Menú principal”.
- Módulo “Personalización”.
- Módulo “Avisos”.

El módulo dispone de dos relaciones de salida de gestión. El resto de relaciones se dirigen a módulos funcionales:

- Módulo de gestión “Gestión proyectos”.
- Módulo de gestión “Gestión eliminar”.
- Módulo “Personalización”.
- Módulo “Avisos”.
- Módulo “Iteraciones”.
- Módulo “Requisitos/Casos de uso”.

- Módulo “Trazabilidad”.
- Módulo “Entregables”.

3.7.4 Módulo Personalización

El objetivo de este módulo es proporcionar un mecanismo para que el usuario pueda seleccionar qué gráficas aparecerán en la pantalla principal de un proyecto concreto.

Las funciones que realizará son las siguientes:

- Activar gráfica en un proyecto concreto.
- Desactivar gráfica en un proyecto concreto.

El módulo cuenta con dos relaciones de entrada y dos relaciones de salida. Las relaciones de entrada provienen de los módulos siguientes:

- Módulo “Proyectos”.
- Módulo “Menú principal”.

Las relaciones de salida son:

- Módulo de gestión “Gestión personalizar”.
- Módulo “Proyectos”.

3.7.5 Módulo Ayuda

El objetivo de este módulo es proporcionar al usuario un manual con toda la información sobre las acciones que puede realizar y la forma de llevarlas a cabo. Su función principal será mostrar el contenido del manual de usuario.

Este módulo dispone únicamente de una relación de entrada con el módulo “Menú principal”. No dispone de ninguna relación de salida.

3.7.6 Módulo Avisos

Este módulo tiene dos objetivos principales:

- Proporcionar al usuario el listado de avisos propios.
- Proporcionar las acciones permitidas sobre un aviso en concreto.

Las funciones que cumplirá son las siguientes:

- Listar los avisos del usuario.
- Leer un aviso concreto.
- Responder un aviso existente.
- Eliminar un aviso existente.

- Crear un nuevo aviso.

Este módulo cuenta con dos relaciones de entrada. Son las siguientes:

- Módulo “Menú principal”.
- Módulo “Proyectos”.

Por otra parte, el módulo cuenta con tres relaciones de salida:

- Módulo de gestión “Gestión avisos”.
- Módulo de gestión “Gestión eliminar”.
- Módulo “Proyectos”.

3.7.7 Módulo Iteraciones

El módulo “Iteraciones” tiene como principal objetivo proporcionar un mecanismo para la gestión de un Plan de Iteraciones para un proyecto en concreto.

Las funciones que cumplirá son las siguientes:

- Listar las iteraciones creadas, clasificadas por fase.
- Crear nueva iteración.
- Editar una iteración existente.
- Eliminar una iteración existente.
- Mostrar los productos pertenecientes a una iteración.
- Ver el historial de una iteración.
- Avanzar a la siguiente fase del proyecto.
- Avanzar a la siguiente iteración de la fase actual y del proyecto.

El módulo dispone de una relación de entrada con el módulo “Proyectos”. Por otra parte, dispone de dos relaciones de salida con los siguientes módulos:

- Módulo de gestión “Gestión eliminar”.
- Módulo de gestión “Gestión iteraciones”.

3.7.8 Módulo Requisitos/Casos de uso

Este módulo tiene como objetivo principal proporcionar un mecanismo para la gestión de requisitos suplementarios y casos de uso en un proyecto concreto.

Las funciones que debe cumplir son las siguientes:

- Listar los requisitos existentes.
- Listar los casos de uso existentes.
- Crear un nuevo requisito.
- Crear un nuevo caso de uso.
- Editar un requisito existente.
- Editar un caso de uso existente.
- Eliminar un requisito existente.
- Eliminar un caso de uso existente.
- Publicar un requisito al Project Manager o al Propietario.
- Publicar un caso de uso al Project Manager o al Propietario.
- Ver el historial de un requisito.
- Ver el historial de un caso de uso.
- Aceptar un requisito publicado.
- Aceptar un caso de uso publicado.
- Rechazar un requisito publicado.
- Rechazar un caso de uso publicado.
- Listar la trazabilidad de un requisito con los casos de uso de los que dependa.

El módulo dispone de una relación de entrada con el módulo “Proyectos”. En cuanto a las relaciones de salida, dispone de las siguientes:

- Módulo “Trazabilidad”.
- Módulo de gestión “Gestión requisitos”.
- Módulo de gestión “Gestión eliminar”.
- Módulo de gestión “Gestión publicar”.

3.7.9 Módulo Trazabilidad

Este módulo tiene dos objetivos principales:

- Proporcionar una matriz de trazabilidad entre requisitos y casos de uso.
- Proporcionar el mecanismo para editar la matriz de trazabilidad.

Las funciones que deberá realizar son las siguientes:

- Crear relación de trazabilidad.
- Eliminar relación de trazabilidad.

El módulo dispone de una relación de salida con el módulo de gestión “Gestión trazabilidad”. Por otra parte, cuenta con dos relaciones de entrada:

- Módulo “Proyectos”.
- Módulo “Requisitos/Casos de uso”.

3.7.10 Módulo Entregables

El objetivo principal del módulo es proporcionar un mecanismo para la gestión de entregables en un proyecto concreto.

Las funciones que deberá cumplir son las siguientes:

- Listar los entregables existentes.
- Crear un nuevo entregable.
- Actualizar un entregable existente.
- Eliminar un entregable existente.
- Publicar un entregable al Project Manager o al Propietario.
- Aceptar un entregable publicado.
- Rechazar un entregable publicado.
- Descargar un entregable existente.

El módulo dispone de una relación de entrada con el módulo “Proyectos”. En cuanto a las relaciones de salida, el módulo “Entregables” cuenta con las siguientes:

- Módulo de gestión “Gestión entregables”.
- Módulo de gestión “Gestión eliminar”.
- Módulo de gestión “Gestión publicar”.

3.7.11 Módulo Gestión Personalizar

Este módulo tiene dos objetivos principales:

- Realizar las funciones del módulo “Personalización”.
- Devolver el resultado de las acciones realizadas al módulo de origen.

Las funciones que deberá realizar son:

- Crear registro en la base de datos con la gráfica y proyecto indicados.
- Eliminar registro en la base de datos con la gráfica y proyecto indicados.

El módulo cuenta con una única relación de entrada con el módulo “Personalización”. Aunque en el diagrama no se muestra ninguna relación de salida, los módulos de gestión siempre se relacionarán con los módulos de origen para devolver el resultado de la acción realizada.

3.7.12 Módulo Gestión Proyectos

Este módulo tiene dos objetivos principales:

- Realizar las funciones del módulo “Proyectos”.
- Devolver el resultado de las acciones realizadas al módulo de origen.

Las funciones que deberá realizar son:

- Crear registro en la base de datos con los datos del nuevo proyecto.
- Actualizar los datos en la base de datos del proyecto indicado con los datos indicados.
- Crear un registro en la base de datos con el nuevo usuario participante en el proyecto indicado.
- Eliminar un registro en la base de datos con el usuario participante y el proyecto indicados.
- Crear un registro en la base de datos para almacenar el historial de las acciones realizadas.

El módulo cuenta con una única relación de entrada con el módulo “Proyectos”. Aunque en el diagrama no se muestra ninguna relación de salida, los módulos de gestión siempre se relacionarán con los módulos de origen para devolver el resultado de la acción realizada.

3.7.13 Módulo Gestión Avisos

El módulo tiene dos objetivos principales:

- Realizar las funciones del módulo “Avisos”.
- Devolver el resultado de las acciones realizadas al módulo de origen.

Las funciones que deberá realizar son las siguientes:

- Crear registro en la base de datos con los datos del nuevo aviso.
- Acceder a la base de datos para leer los detalles de un aviso.

El módulo cuenta con una única relación de entrada con el módulo “Avisos”. Aunque en el diagrama no se muestra ninguna relación de salida, los módulos de gestión siempre se relacionarán con los módulos de origen para devolver el resultado de la acción realizada.

3.7.14 Módulo Gestión Iteraciones

El módulo asume de dos objetivos principales:

- Realizar las funciones del módulo “Iteraciones”.
- Devolver el resultado de las acciones realizadas al módulo de origen.

Las funciones que realizará son:

- Crear registro en la base de datos con los datos de la nueva iteración.
- Acceder a la base de datos para leer las iteraciones existentes para un proyecto en concreto.
- Actualizar un registro de la base de datos con los nuevos datos de la iteración indicada.
- Crear un registro en la base de datos para almacenar el historial de las acciones realizadas.
- Actualizar un registro de la base de datos con la nueva fase indicada.
- Actualizar un registro de la base de datos con la nueva iteración indicada.

El módulo cuenta con una única relación de entrada con el módulo “Iteraciones”. Aunque en el diagrama no se muestra ninguna relación de salida, los módulos de gestión siempre se relacionarán con los módulos de origen para devolver el resultado de la acción realizada.

3.7.15 Módulo Gestión Requisitos

Este módulo tiene dos objetivos principales:

- Realizar las funciones del módulo “Requisitos/Casos de uso”.
- Devolver el resultado de las acciones realizadas al módulo de origen.

Las funciones que deberá realizar son las siguientes:

- Crear registro en la base de datos con los datos del nuevo requisito o caso de uso.
- Acceder a la base de datos para leer los requisitos y casos de uso existentes para un proyecto en concreto.
- Actualizar un registro de la base de datos con los nuevos datos del requisito o caso de uso indicado.

- Crear un registro en la base de datos para almacenar el historial de las acciones realizadas.
- Actualizar un registro de la base de datos con el nuevo estado del requisito o caso de uso indicado.
- Acceder a los registro de la base de datos que contengan las relaciones de trazabilidad entre requisitos y casos de uso.

El módulo cuenta con una única relación de entrada con el módulo “Requisitos/Casos de uso”. Aunque en el diagrama no se muestra ninguna relación de salida, los módulos de gestión siempre se relacionarán con los módulos de origen para devolver el resultado de la acción realizada.

3.7.16 Módulo Gestión Trazabilidad

El módulo tiene dos objetivos que cumplir:

- Realizar las funciones del módulo “Trazabilidad”.
- Devolver el resultado de las acciones realizadas al módulo de origen.

Las funciones que deberá realizar el módulo son las siguientes:

- Crear registro en la base de datos con el requisito y el caso de uso indicados.
- Eliminar registro en la base de datos con el requisito y el caso de uso indicados.

El módulo cuenta con una única relación de entrada con el módulo “Trazabilidad”. Aunque en el diagrama no se muestra ninguna relación de salida, los módulos de gestión siempre se relacionarán con los módulos de origen para devolver el resultado de la acción realizada.

3.7.17 Módulo Gestión Entregables

El módulo tiene dos objetivos principales:

- Realizar las funciones del módulo “Entregables”.
- Devolver el resultado de las acciones realizadas al módulo de origen.

Las funciones que deberá realizar son las siguientes:

- Crear registro en la base de datos con los datos del nuevo entregable.
- Crear documento de forma dinámica con los datos existentes y con los documentos adjuntados.
- Acceder a la base de datos para leer los entregables existentes para un proyecto en concreto.
- Actualizar un registro de la base de datos con los nuevos datos del entregable indicado.

- Actualizar documento de forma dinámica con los datos existentes y con los documentos adjuntados.
- Crear un registro en la base de datos para almacenar el historial de las acciones realizadas.
- Actualizar un registro de la base de datos con el nuevo estado del entregable indicado.

El módulo cuenta con una única relación de entrada con el módulo “Entregables”. Aunque en el diagrama no se muestra ninguna relación de salida, los módulos de gestión siempre se relacionarán con los módulos de origen para devolver el resultado de la acción realizada.

3.7.18 Módulo Gestión Publicar

Los objetivos principales del módulo son:

- Llevar a cabo el cambio al estado “Revisando” o “Publicado” del elemento indicado. Dicho elemento podrá ser un requisito, caso de uso o entregable.
- Devolver el resultado de las acciones realizadas al módulo de origen.

El módulo deberá realizar las siguientes acciones:

- Actualizar un registro de la base de datos con el nuevo estado del elemento indicado.
 - Si el elemento no ha sido revisado por el Project Manager, el estado nuevo será “Revisando”.
 - Si el elemento ha sido revisado por el Project Manager, el estado nuevo será “Publicado”.
- Crear un registro en la base de datos para almacenar el historial de las acciones realizadas.

El módulo dispone de dos relaciones de entrada:

- Módulo “Requisitos/Casos de uso”.
- Módulo “Entregables”.

Aunque en el diagrama no se muestra ninguna relación de salida, los módulos de gestión siempre se relacionarán con los módulos de origen para devolver el resultado de la acción realizada.

3.7.19 Módulo Gestión Eliminar

El módulo tiene dos objetivos principales:

- Eliminar de la base de datos el elemento indicado, así como todos los elementos que dependan directamente del mismo.

- Devolver el resultado de las acciones realizadas al módulo de origen.

Las funciones que deberá cumplir son las siguientes:

- Eliminar registro de la base de datos con el elemento indicado (iteración, requisito, caso de uso, entregable, aviso o proyecto).
- Crear un registro en la base de datos para almacenar el historial de las acciones realizadas.

Este módulo contiene cinco relaciones de entrada:

- Módulo “Proyectos”.
- Módulo “Avisos”.
- Módulo “Iteraciones”.
- Módulo “Requisitos/Casos de uso”.
- Módulo “Entregables”.

Aunque en el diagrama no se muestra ninguna relación de salida, los módulos de gestión siempre se relacionarán con los módulos de origen para devolver el resultado de la acción realizada.

3.8. Disposición de las páginas

En la sección anterior se han mostrado todos los módulos que darán forma a la aplicación web. Estos módulos, exceptuando los módulos de gestión, tendrán su representación visual en páginas web, por lo que es necesario indicar un patrón de diseño para los diferentes módulos identificados.

Existirán cinco tipos de páginas diferentes: login, menú principal, listados, principal del proyecto y formularios de alta o edición. A continuación se muestran las figuras que representan el diseño de cada tipo de página junto con los módulos que pertenecen al tipo de página correspondiente. Las figuras han sido realizadas con la herramienta [Gliffy.com](https://gliffy.com) [11].

3.8.1 Diseño de tipo “Login”

Este tipo de disposición se utilizará únicamente para el módulo “Login”.



The diagram shows a login page layout with the following components and callouts:

- 1**: Enlace a la página principal con el texto "Home".
- 2**: Desplegable que mostrará los proyectos a los que pertenece el usuario conectado. No se mostrará si el usuario no ha accedido a la aplicación.
- 3**: Desplegable que mostrará el nombre del usuario. Si el usuario hace clic sobre él, se mostrarán las opciones "Avisos" y "Salir". El desplegable permanecerá oculto si el usuario no se ha identificado.
- 4**: Texto que mostrará la hora actual en formato "hh:mm".
- 5**: Espacio para mostrar avisos del tipo "Credenciales no válidos" o "Se ha agotado el tiempo de sesión".
- 6**: Espacio para incluir el logotipo de la aplicación.

The layout includes a header with navigation links, a main content area with login fields (Nombre, Contraseña), an "Acceder" button, and a footer area for notices and the application logo.

Figura 18. Diseño de tipo “Login”.

3.8.2 Diseño de tipo “Menú principal”

Este tipo de disposición se utilizará únicamente para el módulo “Menú principal”.

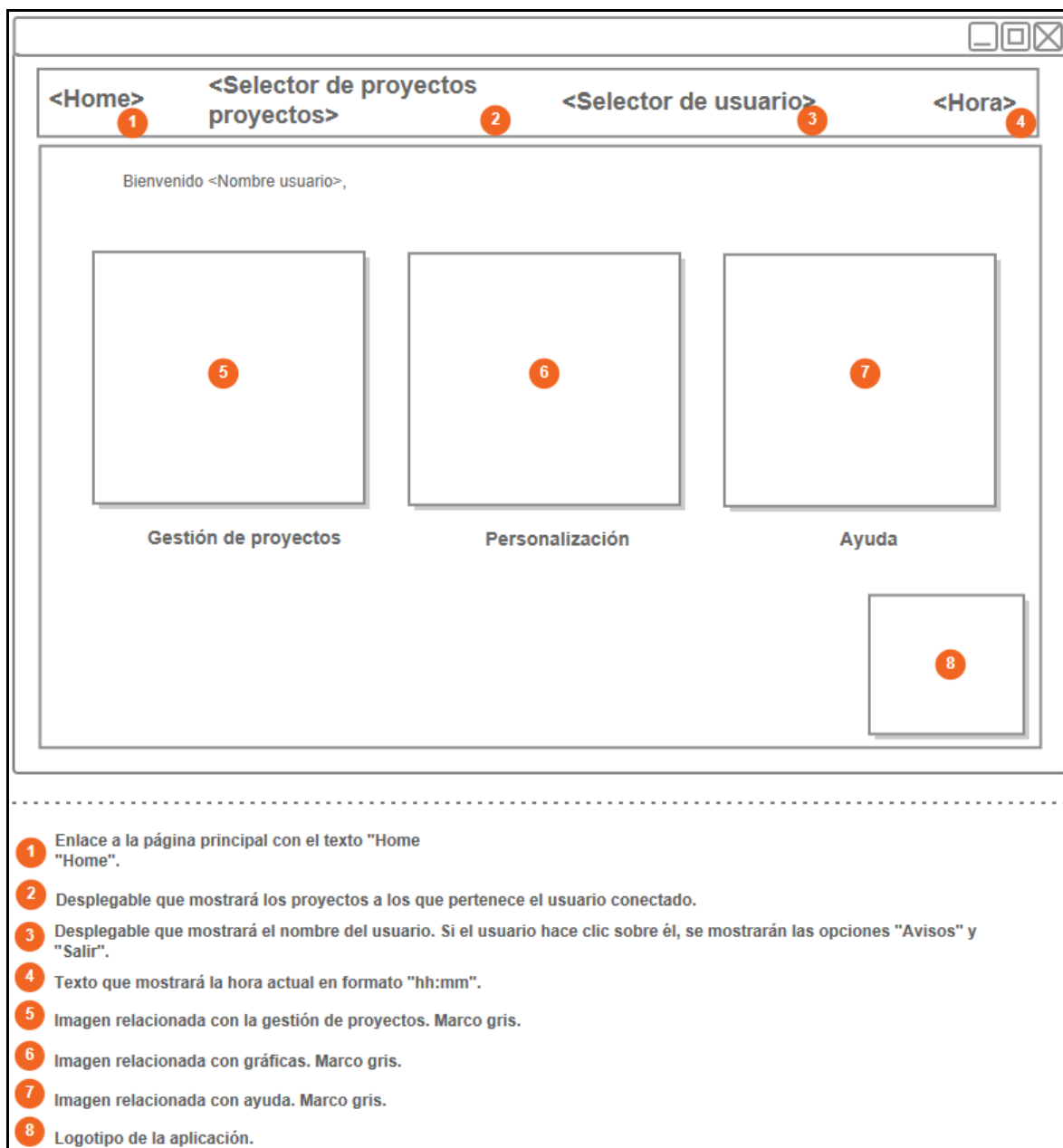


Figura 19. Diseño de tipo “Menú principal”.

3.8.3 Diseño de tipo “Listados”

Este tipo de disposición se utilizará en todos aquellos módulos que listen elementos de un proyecto. Los módulos son:

- Módulo “Proyectos”,
- Módulo “Avisos”,
- Módulo “Iteraciones”,
- Módulo “Requisitos/Casos de uso”,
- Módulo “Entregables”.

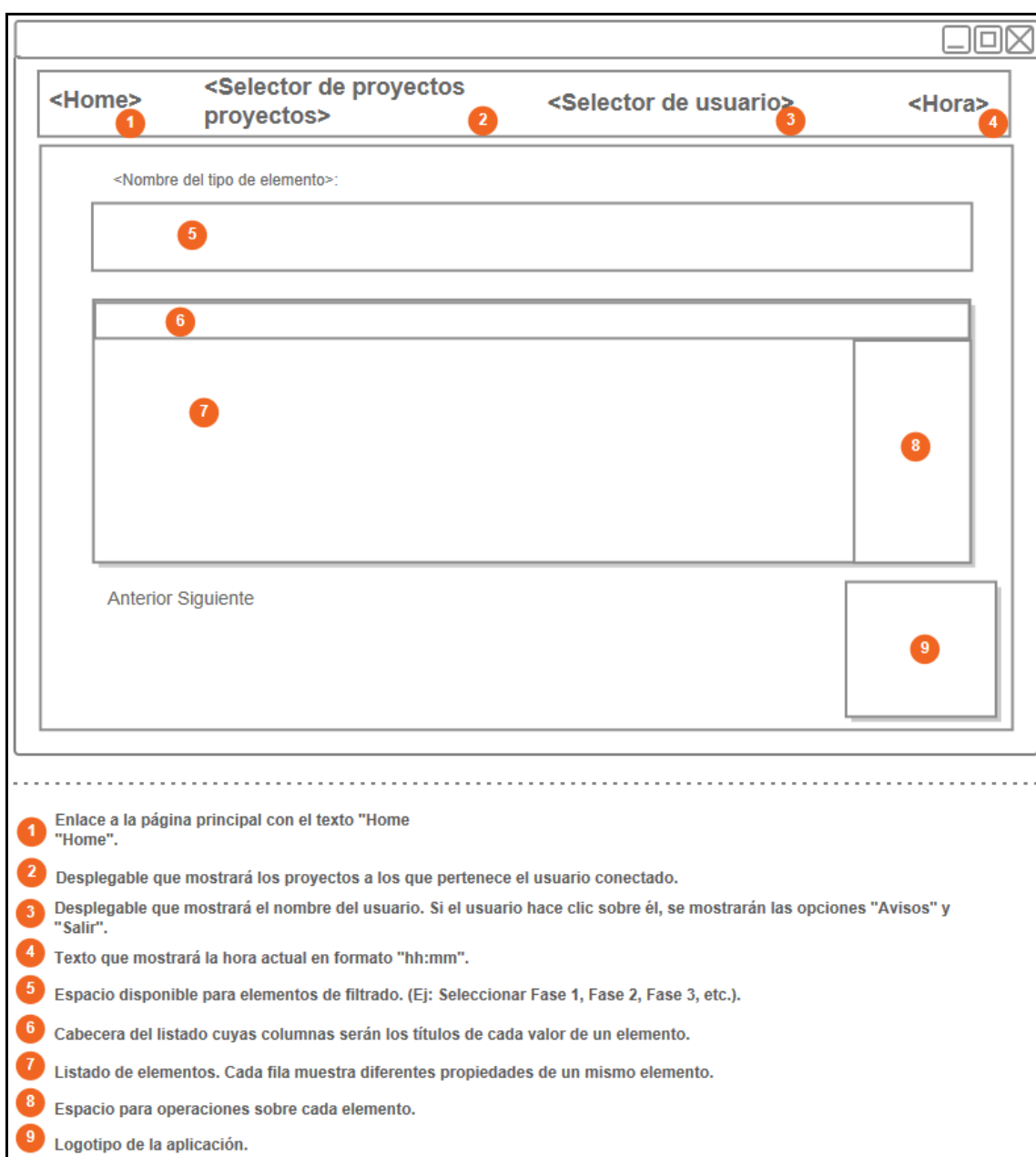


Figura 20. Diseño de tipo “Listados”.

3.8.4 Diseño de tipo “Formularios”

Este tipo de disposición se utilizará en todos aquellos módulos que muestren un formulario con el que interaccionará el usuario, ya sea para dar de alta nuevos elementos o para editar los existentes. Los módulos son:

- Módulo “Proyectos”,
- Módulo “Avisos”,
- Módulo “Iteraciones”,
- Módulo “Requisitos/Casos de uso”,
- Módulo “Entregables”,
- Módulo “Personalización”,
- Módulo “Trazabilidad”.

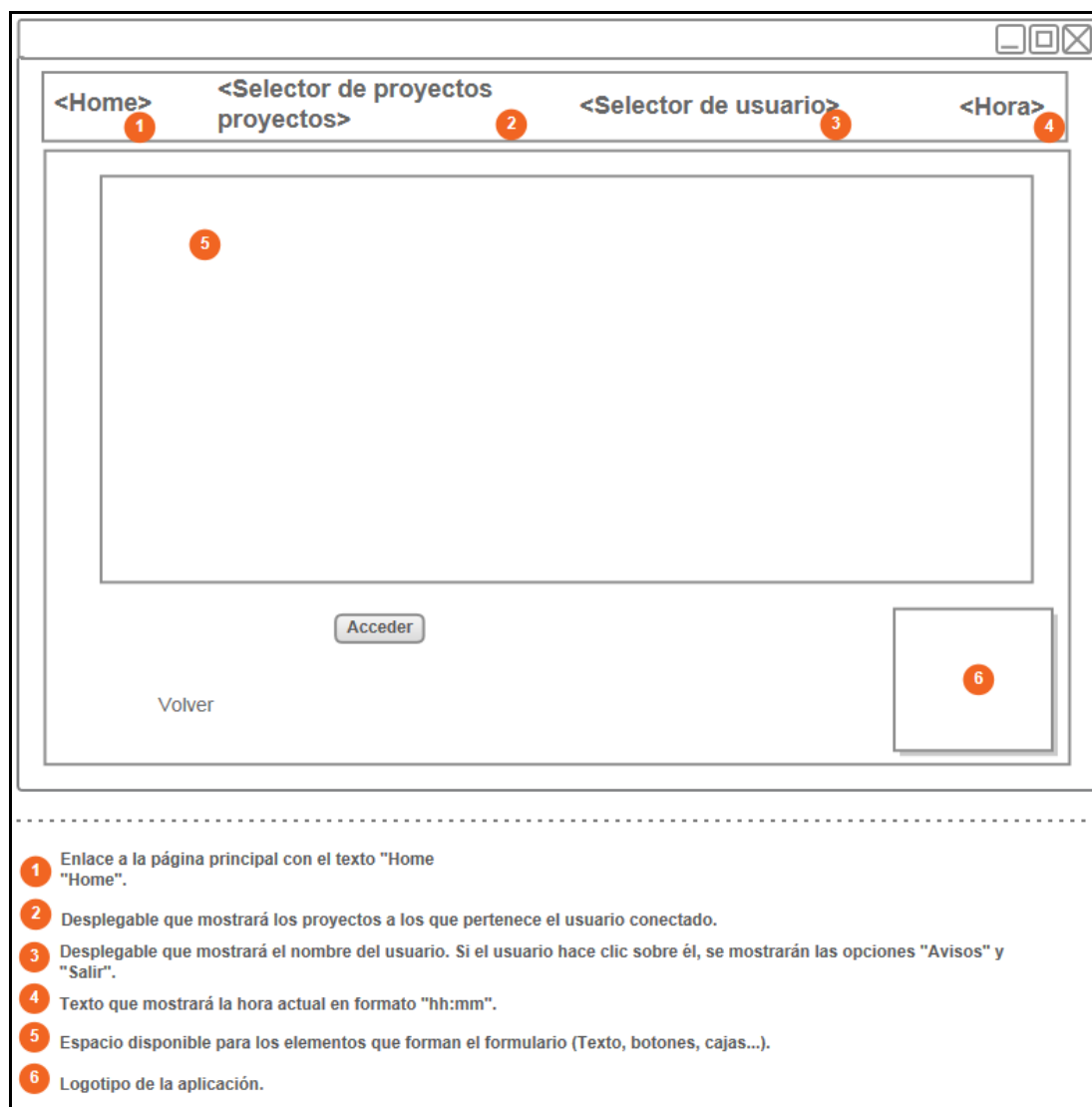


Figura 21. Diseño de tipo “Formularios”.

3.8.5 Diseño de tipo “Principal de un proyecto”

Este tipo de disposición se utilizará únicamente en el módulo “Proyectos”, a la hora de mostrar la página principal del proyecto seleccionado.

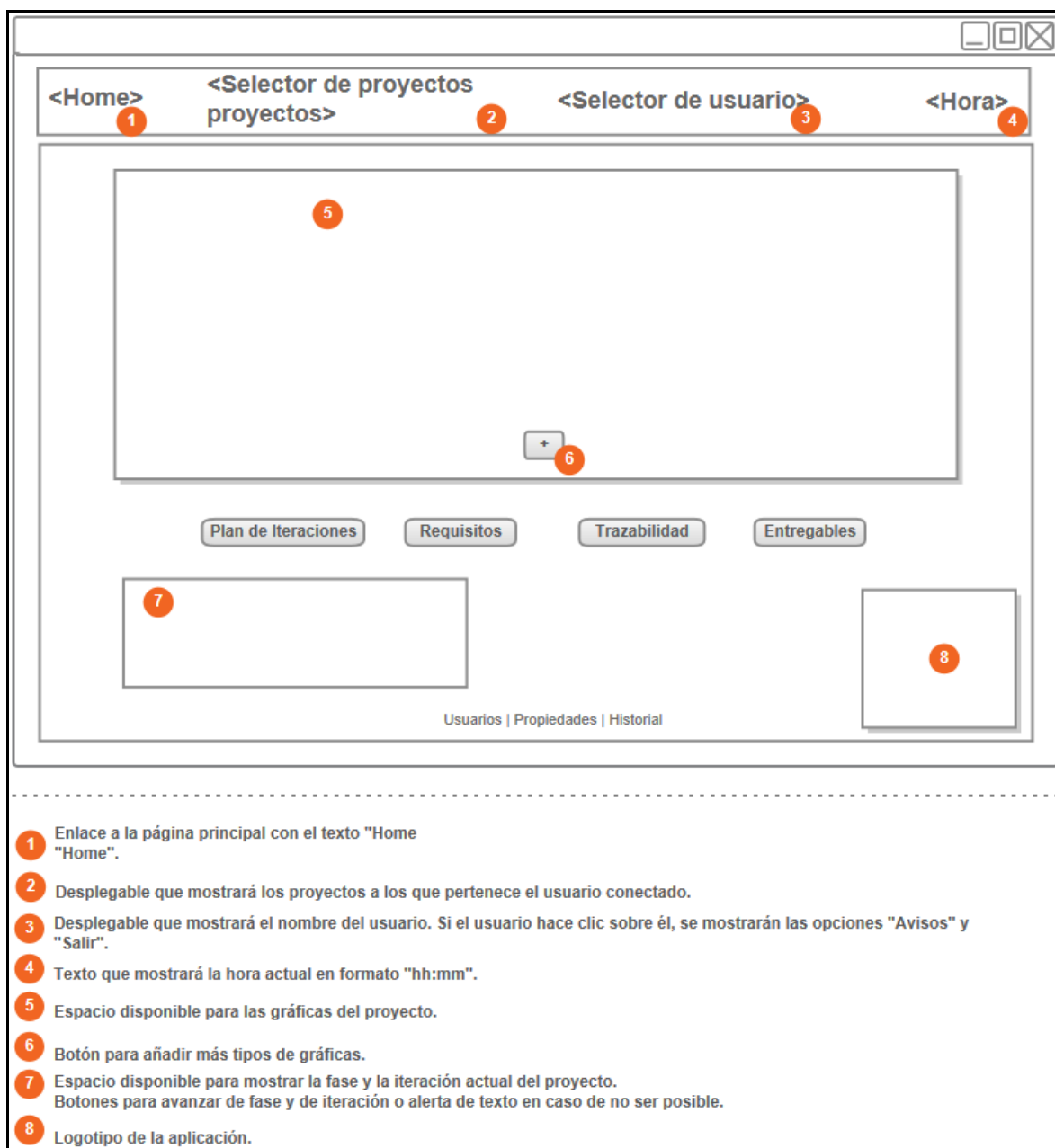


Figura 22. Diseño de tipo “Principal de un proyecto”.

A continuación se muestra el diagrama relacional de la estructura de la base de datos del proyecto. Dicho diagrama ha sido generado con la herramienta [DBSchema](#).



A continuación se especifica una breve explicación para cada entidad, prestando especial interés en las relaciones entre las mismas.

3.9.1 Entidad Proyectos

Es una de las entidades más importantes del sistema. Esta entidad almacenará todos los proyectos que hayan sido creados y servirá de referencia para el resto de entidades. Los atributos y relaciones de los que dispondrá son los siguientes:

ATRIBUTO	DESCRIPCIÓN	FORMATO
id_proyecto	Clave principal de la entidad.	Número entero auto-incremental de 20 bytes.
titulo	Título o nombre del proyecto.	Cadena de caracteres de 100 bytes.
descripcion	Descripción del proyecto.	Cadena de caracteres de 500 bytes.
id_fase	Identificador de la fase actual del proyecto. Será una clave ajena de la entidad “Fases”.	Número entero de 11 bytes.
id_it	Identificador de la iteración actual del proyecto. Será una clave ajena de la entidad “Iteraciones”.	Número entero de 11 bytes.

Tabla 92. Entidad Proyectos.

3.9.2 Entidad Usuarios

Esta entidad se corresponde con la interfaz externa al sistema que será proporcionada junto a la entidad “Roles”. La entidad “Usuarios” proporcionará los datos necesarios de los usuarios que podrán acceder al sistema.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id_usuario	Clave principal de la entidad.	Número entero auto-incremental de 20 bytes.
nombre	Nombre del usuario.	Cadena de caracteres de 50 bytes.
apellidos	Apellidos del usuario.	Cadena de caracteres de 100 bytes.
id_rol	Identificador del rol genérico del usuario. Será una clave ajena de la entidad “Roles”.	Número entero de 20 bytes.
user	Nombre de usuario con el que se podrá identificar en el sistema.	Cadena de caracteres de 20 bytes.
password	Contraseña con la que el usuario podrá acceder al sistema. La contraseña se proporcionará cifrada en MD5 .	Cadena de caracteres de 20 bytes.

Tabla 93. Entidad Usuarios.

3.9.3 Entidad Roles

La entidad “Roles” se corresponde con la segunda interfaz externa. La entidad proporcionará los roles existentes de tal forma que la aplicación clasifique las acciones y permisos dependiendo del rol de cada usuario.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id_rol	Clave principal de la entidad.	Número entero auto-incremental de 20 bytes.
nombre	Nombre del rol.	Cadena de caracteres de 50 bytes.
descripcion	Descripción del rol.	Cadena de caracteres de 500 bytes.

Tabla 94. Entidad Roles.

3.9.4 Entidad Usuarios_participantes

Esta entidad es muy importante, ya que será la que mantenga la relación entre los proyectos creados y los usuarios que pertenezcan a cada proyecto. De igual forma, indicará el rol específico de cada usuario en cada proyecto al que pertenezca.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id_proyecto	Clave principal de la entidad. Clave ajena de la entidad “Proyectos”.	Número entero de 20 bytes.
id_usuario	Clave principal de la entidad. Clave ajena de la entidad “Usuarios”.	Número entero de 20 bytes.
id_rol	Identificador del rol. Clave ajena de la entidad “Roles”.	Número entero de 20 bytes.

Tabla 95. Entidad Usuarios_participantes.

3.9.5 Entidad Historico_participantes

La entidad almacenará un histórico con las acciones realizadas sobre un proyecto concreto.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id_historico	Clave principal de la entidad.	Número entero auto-incremental de 20 bytes.
id_usuario_accion	Identificador del usuario que realizó la acción registrada. Clave ajena de la entidad “Usuarios”.	Número entero de 20 bytes.
id_proyecto	Identificador del proyecto sobre el que se realizó la acción registrada. Clave ajena de la entidad	Número entero de 20 bytes.

	“Proyectos”.	
acción_aplicada_a	Nombre del elemento sobre el que se ha realizado la acción.	Cadena de caracteres de 100 bytes.
accion	Acción que se ha realizado sobre el elemento anterior (crear, editar, eliminar, etc.).	Cadena de caracteres de 100 bytes.
fecha_accion	Fecha en la que se llevó a cabo la acción anterior.	Fecha (Datetime).

Tabla 96. Entidad Historico_participantes.

3.9.6 Entidad Graficas

La entidad almacenará un registro por cada gráfica que los usuarios podrán mostrar en sus proyectos.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id_grafica	Clave principal de la entidad.	Número entero auto-incremental de 20 bytes.
titulo	Nombre descriptivo de la gráfica.	Cadena de caracteres de 100 bytes.

Tabla 97. Entidad Graficas.

3.9.7 Entidad Personalizacion

La entidad será la responsable de relacionar por cada proyecto y cada usuario, las gráficas que éste ha activado.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id_usuario	Clave principal de la entidad. Clave ajena de la entidad “Usuarios”.	Número entero de 20 bytes.
id_proyecto	Clave principal de la entidad. Clave ajena de la entidad “Proyectos”.	Número entero de 20 bytes.
id_grafica	Clave principal de la entidad. Clave ajena de la entidad “Graficas”.	Número entero de 20 bytes.

Tabla 98. Entidad Personalizacion.

3.9.8 Entidad Avisos

La entidad almacenará todos los avisos de todos los usuarios del sistema.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id_aviso	Clave principal de la entidad.	Número entero auto-incremental de 20 bytes.
id_emisor	Identificador del usuario emisor del aviso. Clave ajena de la entidad “Usuarios”.	Número entero de 20 bytes.
id_receptor	Identificador del usuario receptor del aviso. Clave ajena de la entidad “Usuarios”.	Número entero de 20 bytes.
id_proyecto	Identificador del proyecto sobre el que trata el aviso.	Número entero de 20 bytes.
mensaje	El cuerpo del aviso.	Cadena de caracteres de 500 bytes.
fecha	La fecha en la que se creó el aviso.	Fecha (Datetime).
leído	Control que permitirá conocer si el aviso ha sido leído por su receptor.	Cadena de caracteres de 4 bytes.

Tabla 99. Entidad Avisos.

3.9.9 Entidad Iteraciones

Esta entidad será la encargada de almacenar el Plan de Iteraciones de cada proyecto existente.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id_proyecto	Clave principal de la entidad. Clave ajena de la entidad “Proyectos”.	Número entero auto-incremental de 20 bytes.
id_fase	Clave principal de la entidad. Clave ajena de la entidad “Fases”.	Número entero auto-incremental de 11 bytes.
id_it	Clave principal de la entidad. Identificador de la iteración.	Número entero auto-incremental de 11 bytes.
nombre	Nombre de la iteración.	Cadena de caracteres de 50 bytes.
duracion	Duración de la iteración (en semanas).	Número entero de 4 bytes.
estado	El estado de la iteración.	Cadena de caracteres de 20 bytes.

Tabla 100. Entidad Iteraciones.

3.9.10 Entidad Fases

Entidad que almacenará las fases de las que dispondrá un proyecto.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id_fase	Clave principal de la entidad.	Número entero auto-incremental de 11 bytes.
nombre	Nombre de la fase.	Cadena de caracteres de 20 bytes.

Tabla 101. Entidad Fases.

3.9.11 Entidad Productos

Entidad que almacenará los productos que se podrán generar en los proyectos.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id_producto	Clave principal de la entidad.	Número entero auto-incremental de 20 bytes.
nombre	Nombre del producto.	Cadena de caracteres de 20 bytes.

Tabla 102. Entidad Productos.

3.9.12 Entidad Iteracion_producto

Entidad que almacenará los productos que se esperan obtener al finalizar una iteración.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id_proyecto	Clave principal de la entidad. Clave ajena de la entidad "Proyectos".	Número entero de 20 bytes.
id_fase	Clave principal de la entidad. Clave ajena de la entidad "Fases".	Número entero de 11 bytes.
id_it	Clave principal de la entidad. Clave ajena de la entidad "Iteraciones".	Número entero de 11 bytes.
id_producto	Clave principal de la entidad. Clave ajena de la entidad "Productos".	Número entero de 20 bytes.
id_usuario	Identificador del usuario responsable del producto a obtener. Clave ajena de la entidad "Usuarios".	Número entero de 20 bytes.

Tabla 103. Entidad Iteracion_producto.

3.9.13 Entidad Historico_iteraciones

Entidad que almacenará un registro sobre las acciones realizadas sobre una iteración.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id_hist_it	Clave principal de la entidad.	Número entero auto-incremental de 20 bytes.
id_proyecto	Identificador del proyecto al que pertenece la iteración. Clave ajena de la entidad "Proyectos".	Número entero de 20 bytes.
id_fase	Identificador de la fase en la que pertenece la iteración. Clave ajena de la entidad "Fases".	Número entero de 11 bytes.
id_it	Identificador de la iteración de la que se guarda el historial. Clave ajena de la entidad "Iteraciones".	Número entero de 11 bytes.
id_usuario	Identificador del usuario que ha realizado la acción sobre la iteración. Clave ajena de la entidad "Usuarios".	Número entero de 20 bytes.
accion	Acción realizada sobre la iteración.	Cadena de caracteres de 100 bytes.
acción_aplicada_a	Nombre del elemento sobre el que se ha realizado la acción.	Cadena de caracteres de 100 bytes.
valor_anterior	Valor anterior del elemento modificado.	Cadena de caracteres de 100 bytes.
valor_nuevo	Valor nuevo del elemento modificado.	Cadena de caracteres de 100 bytes.
fecha	Fecha de la acción realizada.	Fecha (Datetime).

Tabla 104. Entidad Historico_iteraciones.

3.9.14 Entidad Entregables

Entidad que almacenará todos los datos de los entregables existentes en el sistema.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id_entregable	Clave principal de la entidad.	Número entero auto-incremental de 20 bytes.
id_proyecto	Clave principal de la entidad. Identificador del proyecto al que pertenece el entregable. Clave ajena de la entidad "Proyectos".	Número entero auto-incremental de 20 bytes.
título	Nombre o título del entregable.	Cadena de caracteres de 20 bytes.

descripcion	Descripción del entregable.	Cadena de caracteres de 500 bytes.
ruta	Ruta que apunta al archivo descargable asociado.	Cadena de caracteres de 500 bytes.
estado	Estado actual del entregable.	Cadena de caracteres de 20 bytes.
version	Versión actual del entregable.	Número entero de 4 bytes.
id_fase	Identificador de la fase en la que se creó el entregable. Clave ajena de la entidad "Fases".	Número entero de 11 bytes.
id_it	Identificador de la iteración en la que se creó el entregable. Clave ajena de la entidad "Iteraciones".	Número entero de 11 bytes.

Tabla 105. Entidad Entregables.

3.9.15 Entidad Requisitos

Entidad que almacenará todos los datos de los requisitos existentes en el sistema.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id_requisito	Clave principal de la entidad.	Número entero auto-incremental de 20 bytes.
id_proyecto	Clave principal de la entidad. Identificador del proyecto al que pertenece el requisito. Clave ajena de la entidad "Proyectos".	Número entero auto-incremental de 20 bytes.
titulo	Nombre o título del requisito.	Cadena de caracteres de 20 bytes.
descripcion	Descripción del requisito.	Cadena de caracteres de 500 bytes.
tipo	Tipo de requisito.	Cadena de caracteres de 20 bytes.
subtipo	Subtipo de requisito.	Cadena de caracteres de 20 bytes.
impacto	Impacto del requisito.	Cadena de caracteres de 10 bytes.
esfuerzo	Esfuerzo del requisito.	Cadena de caracteres de 10 bytes.
estado	Estado del requisito.	Cadena de caracteres de 20 bytes.
trazabilidad	Tipo de trazabilidad del requisito.	Cadena de caracteres de 20 bytes.
version	Versión actual del requisito	Número entero de 4 bytes.

Tabla 106. Entidad Requisitos.

3.9.16 Entidad Casos_uso

Entidad que almacenará todos los datos de los casos de uso existentes en el sistema.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id_caso	Clave principal de la entidad.	Número entero auto-incremental de 20 bytes.
id_proyecto	Clave principal de la entidad. Identificador del proyecto al que pertenece el caso de uso. Clave ajena de la entidad "Proyectos".	Número entero auto-incremental de 20 bytes.
titulo	Nombre o título del caso de uso.	Cadena de caracteres de 20 bytes.
ruta_img	Ruta a la imagen que contiene el diagrama del caso de uso.	Cadena de caracteres de 500 bytes.
estado	Estado del caso de uso.	Cadena de caracteres de 20 bytes.
version	Versión actual del caso de uso.	Número entero de 4 bytes.

Tabla 107. Entidad Casos_uso.

3.9.17 Entidad Trazabilidad

Entidad que almacena la trazabilidad entre un requisito y un caso de uso.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id_caso	Clave principal de la entidad. Identificador del caso de uso. Clave ajena de la entidad "Casos_uso".	Número entero de 20 bytes.
id_requisito	Clave principal de la entidad. Identificador del requisito. Clave ajena de la entidad "Requisitos".	Número entero de 20 bytes.

Tabla 108. Entidad Trazabilidad.

3.9.18 Entidad Historico_requisitos

Entidad que almacenará un registro sobre las acciones realizadas sobre un requisito.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id_hist_requisitos	Clave principal de la entidad.	Número entero auto-incremental de 20 bytes.
id_proyecto	Identificador del proyecto al que pertenece el requisito. Clave ajena de la entidad "Proyectos".	Número entero de 20 bytes.

id_requisito	Identificador del requisito al que se la ha aplicado la acción. Clave ajena de la entidad “Requisitos”.	Número entero de 20 bytes.
estabilidad	Estabilidad del requisito.	Cadena de caracteres de 20 bytes.
estado	Estado del requisito.	Cadena de caracteres de 20 bytes.
id_usuario	Identificador del usuario que ha realizado la acción sobre el requisito. Clave ajena de la entidad “Usuarios”.	Número entero auto-incremental de 20 bytes.
accion	Acción realizada sobre el requisito.	Cadena de caracteres de 100 bytes.
acción_aplicada_a	Nombre del elemento sobre el que se ha realizado la acción.	Cadena de caracteres de 100 bytes.
valor_anterior	Valor anterior del elemento modificado.	Cadena de caracteres de 100 bytes.
valor_nuevo	Valor nuevo del elemento modificado.	Cadena de caracteres de 100 bytes.
tipo	Tipo del requisito.	Cadena de caracteres de 20 bytes.
subtipo	Subtipo del requisito.	Cadena de caracteres de 20 bytes.
fecha	Fecha de la acción realizada.	Fecha (Datetime).

Tabla 109. Entidad Historico_requisitos.

3.9.19 Entidad Historico_casos_uso

Entidad que almacenará un registro sobre las acciones realizadas sobre un caso de uso.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id_hist_casos	Clave principal de la entidad.	Número entero auto-incremental de 20 bytes.
id_proyecto	Identificador del proyecto al que pertenece el caso de uso. Clave ajena de la entidad “Proyectos”.	Número entero de 20 bytes.
id_caso	Identificador del caso de uso al que se la ha aplicado la acción. Clave ajena de la entidad “Casos_uso”.	Número entero de 20 bytes.
id_usuario	Identificador del usuario que ha realizado la acción sobre el caso de uso. Clave ajena de la entidad “Usuarios”.	Número entero auto-incremental de 20 bytes.
accion	Acción realizada sobre el caso de uso.	Cadena de caracteres de 100 bytes.

acción_aplicada_a	Nombre del elemento sobre el que se ha realizado la acción.	Cadena de caracteres de 100 bytes.
valor_anterior	Valor anterior del elemento modificado.	Cadena de caracteres de 100 bytes.
valor_nuevo	Valor nuevo del elemento modificado.	Cadena de caracteres de 100 bytes.
fecha	Fecha de la acción realizada.	Fecha (Datetime).

Tabla 110. Entidad Historico_casos_uso.

3.9.20 Entidad Requisitos_proyecto

Entidad que almacena los nuevos tipos de requisitos creados en un proyecto.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id_tipo	Clave principal de la entidad. Identificador del tipo de requisito. Clave ajena de la entidad "Tipos_requisitos".	Número entero de 20 bytes.
id_proyecto	Clave principal de la entidad. Identificador del proyecto al que pertenece el nuevo tipo de requisito. Clave ajena de la entidad "Proyectos".	Número entero de 20 bytes.

Tabla 111. Entidad Requisitos_proyecto.

3.9.21 Entidad Subrequisitos_proyecto

Entidad que almacena los nuevos subtipos de requisitos creados en un proyecto.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id_subtipo	Clave principal de la entidad. Identificador del subtipo de requisito. Clave ajena de la entidad "Subtipos_requisitos".	Número entero de 20 bytes.
id_proyecto	Clave principal de la entidad. Identificador del proyecto al que pertenece el nuevo subtipo de requisito. Clave ajena de la entidad "Proyectos".	Número entero de 20 bytes.

Tabla 112. Entidad Subrequisitos_proyecto.

3.9.22 Entidad Tipos_requisitos

Entidad que almacena los datos de los tipos de requisito.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id	Clave principal de la entidad.	Número entero auto-incremental de 20 bytes.
nombre	Nombre del tipo de requisito.	Cadena de caracteres de 20 bytes.

Tabla 113. Entidad Tipos_requisitos.

3.9.23 Entidad Subtipos_requisitos

Entidad que almacena los datos de los subtipos de requisito.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id	Clave principal de la entidad.	Número entero auto-incremental de 20 bytes.
nombre	Nombre del subtipo de requisito.	Cadena de caracteres de 20 bytes.
id_tipo	Identificador del tipo al que pertenece el subtipo. Clave ajena de la entidad “Tipos_requisitos”.	Número entero de 20 bytes.

Tabla 114. Entidad Subtipos_requisitos.

3.9.24 Entidad Niveles

Entidad auxiliar que almacena los niveles de medición de los atributos “Impacto” y “Esfuerzo” de cada requisito.

ATRIBUTO	DESCRIPCIÓN	FORMATO
id	Clave principal de la entidad.	Número entero auto-incremental de 20 bytes.
nombre	Nombre del nivel.	Cadena de caracteres de 20 bytes.

Tabla 115. Entidad Niveles.

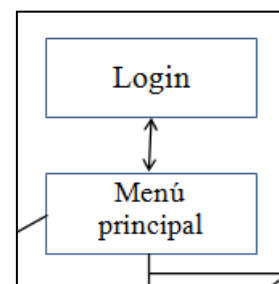
4. Desarrollo

En esta sección se explicará de forma detallada todos los mecanismos utilizados para conseguir el funcionamiento de la aplicación tras su análisis y diseño.

Para mantener una consistencia con la estructura diseñada en la sección “[Diagrama de flujo por bloques](#)”, se partirá de dicha estructura indicando los ficheros creados para realizar la funcionalidad requerida por cada módulo y la relación entre ellos. Con esto se consigue implementar el sistema de forma modular, separando cada funcionalidad de tal forma que sea fácilmente localizable en caso de cambios o actualizaciones.

4.1. Módulo Login

A continuación se especifican los ficheros que componen el módulo “Login”. Por cada fichero se explicará la funcionalidad desarrollada así como la utilización de dicha funcionalidad por parte del usuario final.



4.1.1 install.php

El usuario administrador que lleve a cabo la instalación del programa en su servidor correspondiente, deberá configurar el sistema antes de utilizarlo.

Para ello, se ha creado el fichero install.php cuya finalidad será obtener los datos del servidor para mantener el correcto funcionamiento de la aplicación.

El fichero muestra un formulario HTML al usuario en el que deberá indicar los siguientes datos:

- Nombre de la base de datos donde se gestionarán todos los datos de la aplicación.
- Ruta del servidor MySQL.
- Nombre de usuario del administrador de la base de datos.
- Contraseña del administrador de la base de datos.
- Ruta de instalación de la aplicación.

Bienvenido al instalador

14:41

Siga las siguientes instrucciones para comenzar la instalación.
Asegúrese de que ha copiado los archivos del programa en su servidor antes de comenzar.

1. Configuración de la base de datos.

¡Atención! La base de datos que indicará a continuación debe estar creada en su servidor.

Nombre de la base de datos:

Ruta del servidor MySQL:

Usuario administrador de la base de datos:

Contraseña del usuario anterior:

2. Configuración del servidor.

Para mantener una consistencia con los directorios por defecto, debe indicar en qué ruta ha copiado los datos del programa. **Si los ha copiado en el directorio raíz deje el campo en blanco.**

Ruta de los archivos copiados:

3. Comenzar instalación.

El tiempo estimado de la instalación es de 1 minuto. No haga nada hasta recibir un mensaje de éxito.
Si pasados 2 minutos no ha recibido ningún mensaje o ha recibido un mensaje de error, verifique los datos proporcionados y vuelva a ejecutar el instalable.

Ilustración 1. Instalador de la aplicación.

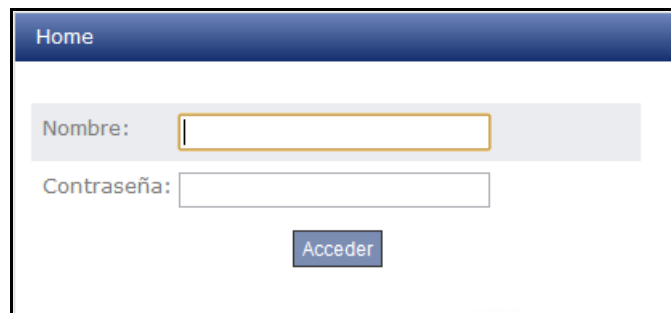
En el momento en que el usuario pulse en el botón “Instalar CPS”, las variables son recogidas a través de [POST](#) por el mismo fichero install.php y comienza la llamada a funciones que realizan la configuración con los datos proporcionados:

- En primer lugar se crea un archivo de configuración (config.php) con los datos anteriores en el directorio “./funciones/”. Este fichero contendrá los datos anteriores en forma de variables para utilizarlas cuando sea necesario. Por motivos de seguridad, todas las variables estarán codificadas en [Base64](#).
- En segundo lugar se vuelca [el script de base de datos](#) (“./sql_init/CPS.sql”) en la base de datos indicada por el usuario.

Si los dos pasos se ejecutan correctamente, se imprime por pantalla un mensaje de éxito y se muestra un enlace para acceder al fichero index.php. En caso contrario, se imprime un mensaje de error.

4.1.2 index.php

El fichero index.php abarca tanto el módulo “Login” como el módulo “Menú principal”.



The screenshot shows a web form titled "Home" in a blue header bar. Below the header, there are two input fields: "Nombre:" with a text box and "Contraseña:" with a password box. Below these fields is a blue button labeled "Acceder".

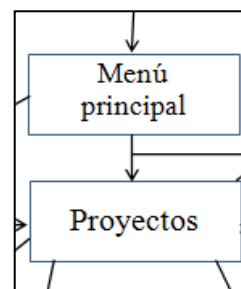
Ilustración 2. Acceso a la aplicación.

El fichero index.php muestra un formulario HTML de acceso a la aplicación con los campos “Usuario” y “Contraseña”. En el momento en que el usuario confirma el formulario, el fichero index.php realiza las siguientes acciones:

- En primer lugar verifica que el usuario proviene del módulo “Login”. Para ello el formulario de acceso dispone de un campo oculto (“hidden”) indicando el módulo donde se encuentra.
- Si es así, se llama a la función “identificarUsuario(usuario, contraseña)”. Esta función se conecta a la base de datos y compara el [MD5](#) de la contraseña indicada junto con el MD5 de la contraseña del usuario almacenada.
- Si la función devuelve true, el fichero muestra un mensaje de error por pantalla. En caso contrario, procede a la asignación de variables de sesión. Entre las variables se encuentra una variable de control (“access”) que permite comprobar si el usuario mantiene una sesión activa, otras variable que almacenan el nombre de usuario, el identificador del usuario en la base de datos, su nombre y apellidos y su rol general.
- Por último, el fichero index.php comprueba el valor de la variable “access”. Si es true, significa que el usuario se ha identificado correctamente en el sistema, por lo que imprime la pantalla correspondiente al módulo “Menú Principal”. En caso contrario, muestra un mensaje de error.

4.2. Módulo Menú principal

A continuación se especifican los ficheros que componen el módulo “Menú principal”. Por cada fichero se explicará la funcionalidad desarrollada así como la utilización de dicha funcionalidad por parte del usuario final.



4.2.1 index.php

En este estado, el usuario ya se ha identificado, por lo que la función principal del fichero index.php consiste en mostrar el menú principal de la aplicación.



Ilustración 3. Menú principal.

Para ello, muestra en la parte central de la pantalla tres opciones: “Gestión de proyectos”, “Personalización de gráficas” y “Ayuda”. Los textos van acompañados de una imagen en forma de enlace, de tal forma que si el usuario selecciona una opción, se muestre la pantalla correspondiente.

Aparte de las tres opciones principales, el fichero index.php imprime una cabecera en la parte superior de la pantalla desde la que el usuario podrá seleccionar los proyectos a los que pertenece, dirigirse al módulo “Avisos” o salir del sistema. Esta cabecera estará visible en todas las pantallas del sistema y por tanto estará implementada en todos los ficheros que contengan interfaz gráfica.

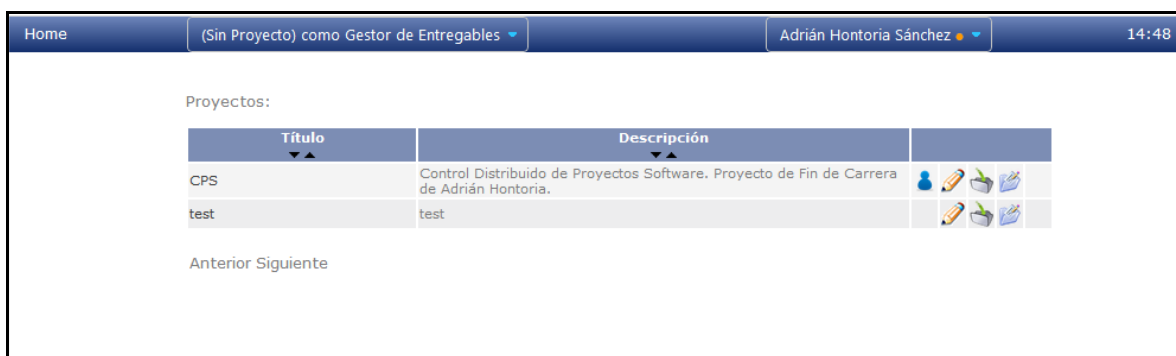
4.3. Módulo Proyectos






A continuación se especifican los ficheros que componen el módulo “Proyectos”. Por cada fichero se explicará la funcionalidad desarrollada así como la utilización de dicha funcionalidad por parte del usuario final.

4.3.1 proyectos.php

El fichero proyectos.php muestra una tabla con los proyectos a los que pertenece el usuario, junto con las acciones que puede realizar sobre él.

Para ello, en primer lugar realiza una consulta a la base de datos utilizando la variable de sesión que almacena el identificador del usuario. De esta forma obtiene los proyectos pertenecientes al usuario conectado.



Título	Descripción	
CPS	Control Distribuido de Proyectos Software. Proyecto de Fin de Carrera de Adrián Hontoria.	  
test	test	 

Anterior Siguiente

Ilustración 4. Listado de proyectos.

Tras imprimir el nombre de cada proyecto, a continuación se imprimen una serie de imágenes relacionadas con la acción a realizar sobre el proyecto situado en la misma fila de la tabla (Editar, eliminar...). Estas imágenes son realmente formularios HTML con campos ocultos, cuya función será realizada por el fichero correspondiente de cada formulario (“action”).

De igual forma, el nombre de los proyectos contiene un enlace a la pantalla principal de cada proyecto (“index_proyecto.php”).

A continuación, el fichero muestra un enlace que permitirá al usuario crear un nuevo proyecto, siempre que cuente con los permisos necesarios. Para ello, el fichero comprueba el rol general almacenado en una variable de sesión y si el usuario es administrador, imprime la opción mencionada. De la misma forma, muestra la opción de asignar participantes a los proyectos existentes.

4.3.2 index_proyecto.php

Este fichero es uno de los más complejos del sistema. Se encarga de mostrar todas las opciones de gestión de elementos internos del proyecto, así como las gráficas para controlar el estado del proyecto.

En primer lugar se obtiene el identificador del proyecto a través de [POST](#) o [GET](#), dependiendo del origen de la llamada al fichero. Una vez obtenido el identificador, se

obtienen todos los datos del proyecto a través de consultas a la base de datos (título, descripción, fase actual, iteración actual...). De igual forma, se obtiene el rol específico del usuario con sesión iniciada en el proyecto actual. Dicho rol se almacena en una nueva variable de sesión. También se almacena en una nueva variable de sesión el proyecto actual. Además dicho proyecto se muestra en primer lugar en el listado de proyectos situado en la cabecera superior.



Ilustración 5. Pantalla principal del proyecto.

A continuación se verifica qué gráficas ha activado el usuario para el proyecto actual a través de una consulta a la base de datos. Así, el fichero hace visibles únicamente las gráficas activadas.

En la parte inferior del área de las gráficas, el fichero muestra los enlaces en forma de botón para acceder a diferentes módulos: “Iteraciones”, “Requisitos/Casos de uso”, “Trazabilidad” y “Entregables”.

Siguiendo la estructura del fichero, a continuación se muestra la fase actual y la iteración actual. Seguido del dato actual, se realiza verifica si está permitido pasar de fase o de iteración. Para ello se realiza una consulta a la base de datos para comprobar si existen iteraciones futuras y si se han generado todos los productos de la iteración actual. Si todo es correcto, se muestra el botón “Pasar de fase” y/o “Pasar de iteración”. En caso contrario, se muestra un mensaje con el estado del proyecto que impide avanzar de fase y/o iteración.

Por último, en la parte inferior de la pantalla se imprimen tres enlaces que conducen a la asignación de participantes del proyecto ([asignarProyecto.php](#)), propiedades del proyecto ([editar_proyecto.php](#)) e historial del proyecto ([historial_proyectos.php](#)).

4.3.3 crear_proyecto.php

Este fichero es muy sencillo. Básicamente muestra un formulario con los campos “Título” y “Descripción”. El usuario deberá indicar ambos datos de forma obligatoria, ya que se ha implementado una función Javascript que mostrará un aviso de error en el momento en el que el usuario confirme el envío del formulario.

Home (Sin Proyecto) como Administrador

Título:

Descripción:

Aceptar

Ilustración 6. Creación de un nuevo proyecto.

Si el usuario ha indicado ambos datos de forma correcta, se llama al módulo “Gestión proyectos” para que lleve a cabo la creación del proyecto en la base de datos.

4.3.4 editar_proyecto.php

El fichero `editar_proyecto.php` es muy similar al fichero `crear_proyecto.php`. La única diferencia reside en el valor de los campos “Título” y “Descripción”. En este caso dichos campos aparecerán con la información del proyecto a editar. Dicha información se obtiene a través de una consulta a la base de datos antes de mostrar el formulario.

Al igual que en el momento de la creación, el usuario deberá indicar ambos datos de forma obligatoria, ya que se ha implementado una función Javascript que mostrará un aviso de error en el momento en el que el usuario confirme el envío del formulario.

Home (Sin Proyecto) como Administrador

Título: CPS

Descripción: Control Distribuido de Proyectos Software. Proyecto de Fin de Carrera de Adrián Hontoria.

Aceptar Volver

Ilustración 7. Edición de un proyecto existente.

Si el usuario ha indicado ambos datos de forma correcta, se llama al módulo “Gestión proyectos” para que lleve a cabo la edición del proyecto en la base de datos.

4.3.5 grafica_bar1.php, grafica_bar2.php y grafica_pie.php

A la hora de generar las gráficas, es necesario contar con ficheros genéricos que construyan las gráficas de forma dinámica dependiendo de los datos que se importen.

Este es el objetivo de los ficheros `grafica_bar1.php`, `grafica_bar2.php` y `grafica_pie.php`. Los ficheros utilizan la librería `jpggraph.php` incluida en el directorio “./funciones/graficas/”. Estos ficheros reciben a través de GET todos los datos de la gráfica a generar (nombre, título de los ejes de coordenadas, los datos a mostrar en forma de [arrays](#),

etc.) y a través de las funciones proporcionadas por la librería `jpggraph.php` se genera al instante la gráfica deseada.

La diferencia entre los tres ficheros generadores reside en el tipo de gráfica:

- `grafica_bar1.php` genera una gráfica de barras tras recibir dos fuentes de datos (ejex y ejey).

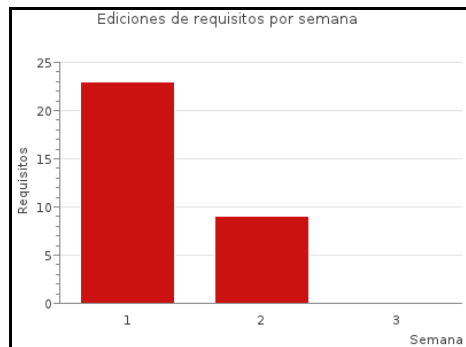


Ilustración 8. Gráfica de barras de tipo 1.

- `grafica_bae2.php` genera una gráfica de barras tras recibir tres fuentes de datos (ejex, ejey y color en cada barra junto a una leyenda).

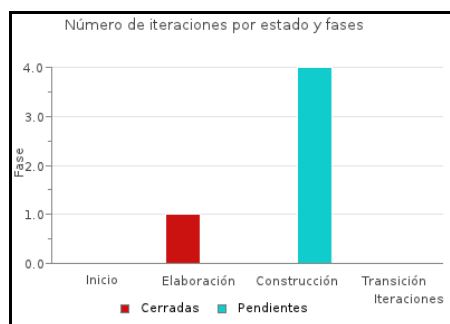


Ilustración 9. Gráfica de barras de tipo 2.

- `grafica_pie.php` genera un gráfico circular tras recibir un único [array](#) de datos.



Ilustración 10. Gráfica circular.

4.3.6 asignarProyecto.php

Este fichero es el responsable de gestionar la asignación de personas y roles a los proyectos.

En primer lugar recibe por parámetro el identificador del proyecto a gestionar. Si la variable está vacía significa que el usuario ha accedido desde la opción “Asignar participantes” desde index.php, por lo que podrá gestionar cualquier proyecto al que pertenezca (siempre que cuente con los permisos necesarios).



Ilustración 11. Asignación de usuarios.

En el caso en el que se puedan gestionar todos los proyectos, el fichero realiza una llamada a la base de datos para obtener el nombre de cada proyecto. Todos estos proyectos se mostrarán en un selector.

El fichero tiene implementada una función de refresco cada vez que el usuario cambie el valor del selector. De esta forma, se actualizarán los usuarios y los roles del proyecto seleccionado de forma dinámica.

Los usuarios dispondrán en la parte izquierda de un control “[checkBox](#)” y en la parte derecha de un selector con los diferentes roles existentes. De esta forma, el usuario que gestione a los participantes, podrá seleccionar qué usuario participará o no participará en el proyecto (activando o desactivando el “checkBox”) y podrá asignar un rol a cada usuario (seleccionando un rol del listado de cada usuario).

Una vez confirmado el formulario, las funciones correspondientes se llevarán a cabo por el módulo “Gestión proyectos”.

4.3.7 gestion_zip.php

Los usuarios podrán descargar todos los entregables generados de un proyecto, comprimidos en [.zip](#) desde el listado de proyectos. Es por ello necesario contar con un mecanismo que acceda a dichos entregables y los comprima en formato .zip.

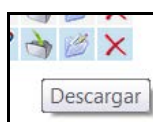


Ilustración 12. Descarga de un proyecto.

De esto se encarga el fichero `gestion_zip.php`. Este fichero se apoya en la librería `zipfile.php` incluida en el directorio “./funciones/”. Esta librería proporciona una clase “`zipfile`” con dos funciones principales:

- `addfile(fichero)`: añade un fichero al futuro `.zip`.
- `file()`: genera el fichero `.zip` con los ficheros añadidos anteriormente.

De esta forma el fichero `gestion_zip.php` recorre todos los entregables generados para el proyecto seleccionado y los va añadiendo con la función “`addfile(entregable)`”. Tras recorrer todos los entregables, se genera el fichero comprimido `.zip` (cuyo nombre será el título del proyecto) y se le ofrecerá al usuario para su descarga.

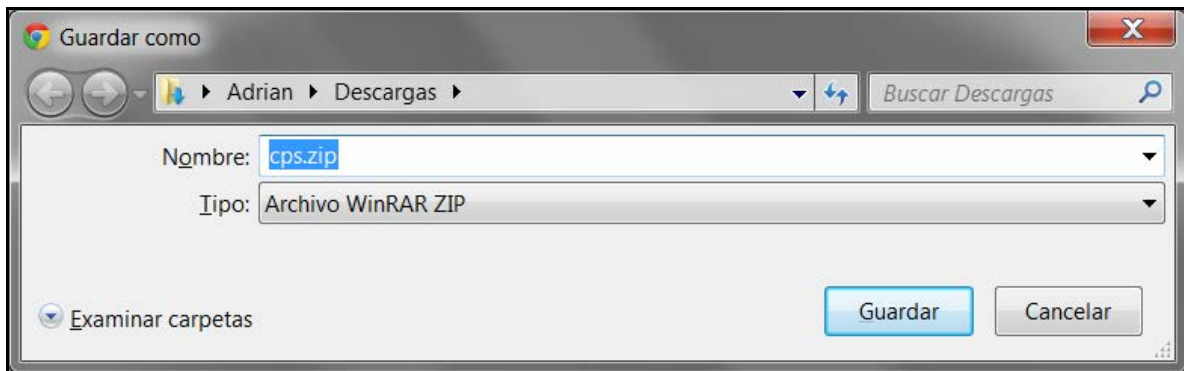


Ilustración 13. Descarga remota de un proyecto.

4.3.8 historial_proyectos.php

Como su nombre indica, este fichero muestra el historial completo del proyecto seleccionado. En primer lugar, obtiene el identificador del proyecto a través de la variable de sesión o bien a través del parámetro recibido.

A continuación, el fichero realiza una consulta múltiple para acceder a todas las acciones realizadas sobre el proyecto.

Historial de CPS:

Realizado por: ▼▲	Acción realizada ▼▲	Acción aplicada a: ▼▲	Fecha ▼▲
Luis Miguel Sánchez García	Rechazar	Requisito Gestionar casos de uso	2012-04-22 06:57:27
José Daniel García Sánchez	Publicar a PM	Requisito Gestionar trazabilidad	2012-04-22 06:56:51
Sr. Administrador Apellido	Alta como Ingeniero de Requisitos	José Daniel García Sánchez	2012-04-22 06:56:32
Sr. Administrador Apellido	Publicar a cliente	Requisito Gestionar casos de uso	2012-04-22 06:55:52
Sr. Administrador Apellido	Publicar a cliente	Requisito Gestionar requisitos	2012-04-22 06:55:47
Sr. Administrador Apellido	Editar requisito	Requisito Gestionar iteraciones	2012-04-22 06:55:36
Adrián Hontoria Sánchez	Actualizar	Entregable	2012-04-09 06:41:27
Adrián Hontoria Sánchez	Editar caso de uso	Caso de uso Gestionar casos de uso	2012-04-09 06:29:25
Adrián Hontoria Sánchez	Publicar a cliente	Entregable Modelo de casos de uso	2012-04-09 06:27:49
Adrián Hontoria Sánchez	Publicar a cliente	Caso de uso Gestionar requisitos	2012-04-09 06:27:12
Adrián Hontoria Sánchez	Publicar a cliente	Caso de uso Gestionar iteraciones	2012-04-09 06:27:00
Adrián Hontoria Sánchez	Publicar a cliente	Caso de uso Gestionar proyectos	2012-04-09 06:26:52
Adrián Hontoria Sánchez	Actualizar	Entregable	2012-03-29 05:20:08
Adrián Hontoria Sánchez	Actualizar	Entregable	2012-03-29 05:20:02
Luis Miguel Sánchez García	Aceptar	Requisito Permiso de Gestor de entregables	2012-03-28 07:52:26

Anterior [Siguiente](#)

[Volver](#)

Ilustración 14. Historial del proyecto.

El fichero muestra una tabla con todos los datos obtenidos de la base de datos. Cada acción se muestra en cada fila de la tabla. El número de filas está limitado a 15 por página. Tras el listado de acciones, el fichero muestra los enlaces “Anterior” y “Siguiente” para navegar por las diferentes páginas del historial.

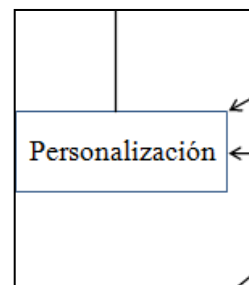
De forma adicional, los registros de la tabla se pueden ordenar por columna y en orden ascendente o descendente, dependiendo de la selección del icono correspondiente.



Ilustración 15. Ordenación de datos.

4.4. Módulo Personalización

A continuación se especifican los ficheros que componen el módulo “Personalización”. Por cada fichero se explicará la funcionalidad desarrollada así como la utilización de dicha funcionalidad por parte del usuario final.



4.4.1 personalizar.php

El fichero personalizar.php muestra en primer lugar un selector de proyectos a los que el usuario pertenece. Al seleccionar alguno de estos proyectos, se cargarán las gráficas que tiene activas para dicho proyecto tras consultarlo a la base de datos.



Ilustración 16. Activación y desactivación de gráficas.

En la parte izquierda del nombre de cada gráfica, el fichero muestra un “checkbox” activado o desactivado dependiendo de si el usuario tiene activada o no la gráfica correspondiente.

El usuario podrá seleccionar qué gráficas activar o desactivar y al pulsar el botón “Aceptar” el fichero enviará los datos al módulo “Gestión personalizar” para llevar a cabo la acción correspondiente.

4.5. Módulo Ayuda

Para la sección de ayuda, se ha recurrido a un servicio externo: “wikispaces.com”. Dicho servicio proporciona una interfaz de ayuda online, en la que se han creado una serie de secciones a las que cualquier usuario podrá recurrir en caso de dudas o problemas con el sistema. Todo el contenido de la ayuda se encuentra en <http://ayudacps.wikispaces.com>.

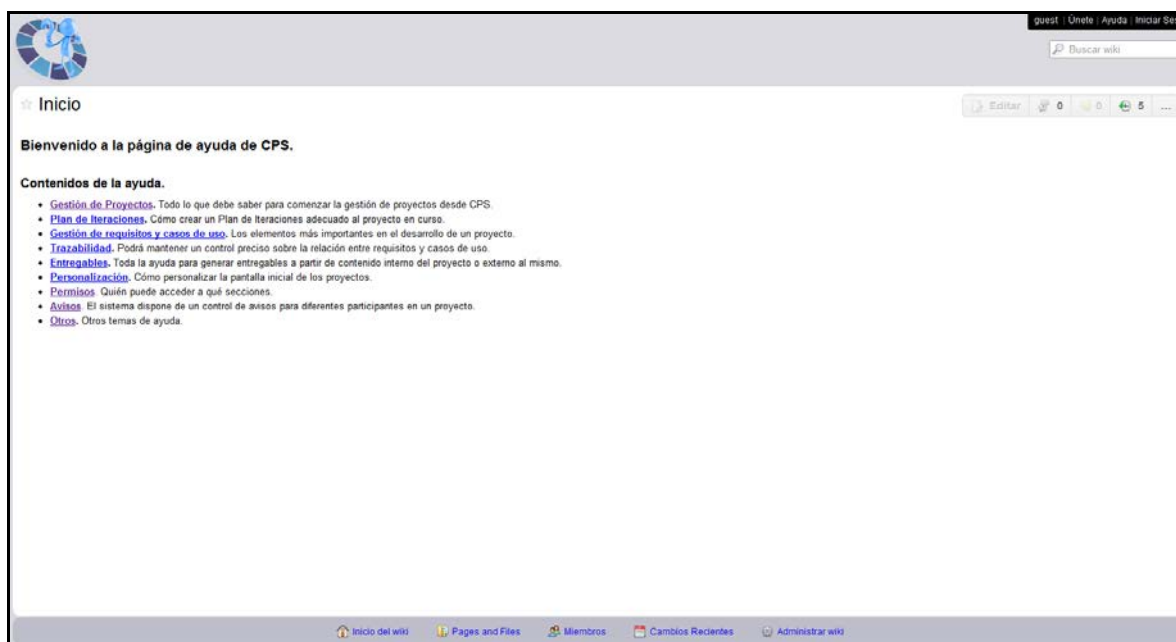
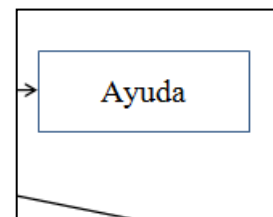
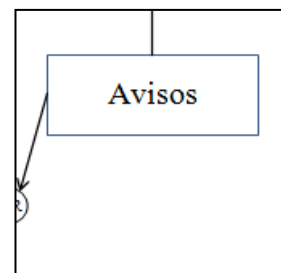


Ilustración 17. Ayuda a los usuarios.

4.6. Módulo Avisos

A continuación se especifican los ficheros que componen el módulo “Avisos”. Por cada fichero se explicará la funcionalidad desarrollada así como la utilización de dicha funcionalidad por parte del usuario final.



4.6.1 avisos.php

La funcionalidad principal del fichero consiste en mostrar el listado de avisos de un usuario. Dicho usuario será el que tenga iniciada la sesión y por tanto el fichero avisos.php accederá a la base de datos indicando el identificador guardado en la variable de sesión correspondiente.

Home	(Sin Proyecto) como Gestor de Entregables	Adrián Hontoria Sánchez	11:52
Nuevo aviso			
Remitente	Proyecto	Mensaje	Fecha
Sr. Administrador Apellido	test	Test	2012-03-06 09:57:35
Luis Miguel Sánchez García	CPS	Requisito aceptado: Gestionar proyectos	2012-03-20 14:54:48
Luis Miguel Sánchez García	CPS	Requisito aceptado: Gestionar iteraciones	2012-03-20 14:54:52
Luis Miguel Sánchez García	CPS	Requisito aceptado: Gestionar requisitos	2012-03-20 14:54:57
Luis Miguel Sánchez García	CPS	Requisito aceptado: Gestionar casos de uso	2012-03-20 14:55:05
Luis Miguel Sánchez García	CPS	Requisito aceptado: Gestionar trazabilidad	2012-03-20 14:55:10
Luis Miguel Sánchez García	CPS	Requisito aceptado: Gestionar entregables	2012-03-20 14:55:14
Luis Miguel Sánchez García	CPS	Requisito aceptado: Datos casos de uso	2012-03-20 14:55:18
Luis Miguel Sánchez García	CPS	Requisito aceptado: Generación de entregables	2012-03-20 14:55:23
Luis Miguel Sánchez García	CPS	Requisito aceptado: Generación manual de entregables	2012-03-20 14:55:27
Luis Miguel Sánchez García	CPS	Requisito aceptado: Generación automática de entregables	2012-03-20 14:55:32
Luis Miguel Sánchez García	CPS	Requisito aceptado: Formato de documentos	2012-03-20 14:55:36
Luis Miguel Sánchez García	CPS	Requisito aceptado: Formato de entregables	2012-03-20 14:55:41
Luis Miguel Sánchez García	CPS	Requisito aceptado: Datos de iteraciones	2012-03-20 14:55:49
Luis Miguel Sánchez García	CPS	Requisito aceptado: Datos de proyectos	2012-03-20 14:55:54
Luis Miguel Sánchez García	CPS	Requisito aceptado: Datos de requisitos	2012-03-20 14:55:59
Luis Miguel Sánchez García	CPS	Requisito aceptado: Tipos de requisitos	2012-03-20 14:56:03
Luis Miguel Sánchez García	CPS	Requisito aceptado: Subtipos de requisitos	2012-03-20 14:56:08
Luis Miguel Sánchez García	CPS	Requisito aceptado: Interfaz de usuarios	2012-03-20 14:56:11
Luis Miguel Sánchez García	CPS	Requisito aceptado: Clasificación de usuarios	2012-03-20 14:56:15
Luis Miguel Sánchez García	CPS	Requisito aceptado: Gestionar participantes en un proyecto	2012-03-20 14:56:20
Luis Miguel Sánchez García	CPS	Requisito aceptado: Listado de proyectos	2012-03-20 14:56:23
Luis Miguel Sánchez García	CPS	Caso de uso aceptado: bld	2012-03-21 10:41:18
Luis Miguel Sánchez García	CPS	Requisito aceptado: Gestionar proyectos	2012-03-27 14:07:39
Luis Miguel Sánchez García	CPS	Requisito aceptado: Gráfica "Entregables por estado"	2012-03-28 07:52:20
Luis Miguel Sánchez García	CPS	Requisito aceptado: Permiso de Gestor de entregables	2012-03-28 07:52:26
Anterior Siguiente			
Volver			

Ilustración 18. Avisos.

Una vez accedidos a los avisos del usuario, se imprimen en una tabla. Por cada fila, se indica un aviso con el emisor del mismo, el mensaje, la fecha y las acciones a realizar para el aviso: responder ([responder_aviso.php](#)) o eliminar ([gestion_eliminar.php](#)). De forma adicional, el nombre del aviso es un enlace al fichero [leer_aviso.php](#).

El fichero también muestra la opción “Nuevo aviso”, que enlaza al fichero [crear_aviso.php](#). Es obligatorio que el usuario haya seleccionado un proyecto para poder crear un aviso. Si no es así, el fichero lanza un mensaje de error comunicando el motivo.

Al fichero avisos.php se llega desde la opción “Avisos”, situada en el segundo desplegable de la cabecera superior.

4.6.2 crear_aviso.php

El fichero crear_aviso.php muestra un formulario simple con dos elementos:

- Un listado (selector) con los participantes del proyecto seleccionado: el usuario seleccionado será el receptor del aviso.
- Un campo de texto donde se incluirá el contenido del aviso.

Ilustración 19. Crear aviso.

La única comunicación de este fichero con la base de datos es la obtención de los participantes del proyecto seleccionado. Para ello, accede a la variable de sesión que contiene el identificador del proyecto y realiza la consulta correspondiente a la base de datos.

La gestión del nuevo aviso, en el momento en que el usuario confirma el formulario, es realizada en el módulo “Gestión avisos”.

4.6.3 leer_aviso.php

Este fichero recibe a través de GET y de forma codificada el identificador del aviso que fue seleccionado desde el fichero avisos.php. El fichero leer_aviso.php decodifica la variable y accede a la base de datos para consultar toda la información del aviso.

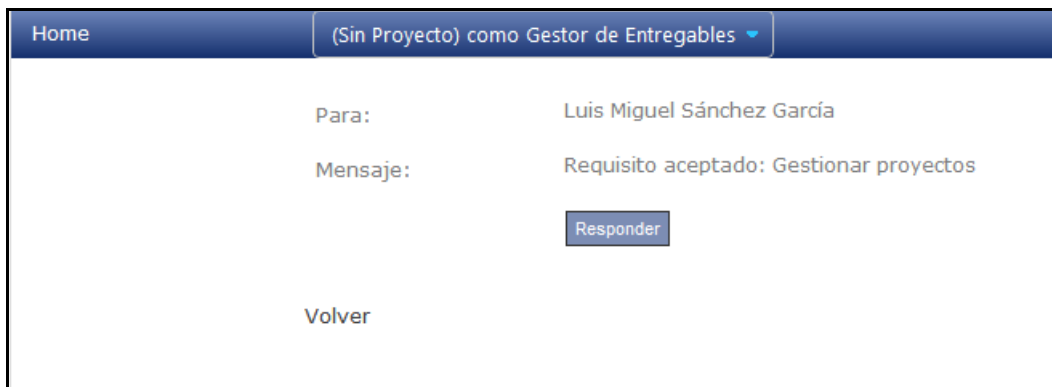


Ilustración 20. Leer aviso.

El fichero, tras recoger los datos del aviso de la base de datos, los imprime por pantalla. Además proporciona un enlace al fichero [responder_aviso.php](#).

4.6.4 responder_aviso.php

Este fichero es muy similar a crear_aviso.php, con la diferencia en que el usuario destinatario es fijo y no puede modificarse.



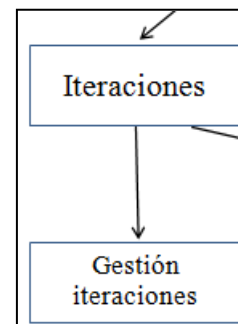
Ilustración 21. Responder aviso.

El fichero recibe por GET y de forma codificada el identificador del aviso a responder. Así, el fichero accede a la base de datos para consultar el emisor del aviso para así mostrarlo por pantalla y añadirlo como destinatario del nuevo aviso a crear.

La confirmación del formulario y por tanto la respuesta del aviso se realiza desde el módulo “[Gestión avisos](#)”.

4.7. Módulo Iteraciones

A continuación se especifican los ficheros que componen el módulo “Iteraciones”. Por cada fichero se explicará la funcionalidad desarrollada así como la utilización de dicha funcionalidad por parte del usuario final.



4.7.1 iteraciones.php

La función principal de este fichero es mostrar las iteraciones creadas, clasificadas por fase, para el proyecto seleccionado.

En primer lugar, el fichero accede al identificador del proyecto seleccionado, que se encuentra almacenado en la variable de sesión correspondiente. Gracias al identificador, el fichero accede a todas las iteraciones creadas en el proyecto. Aparte de recoger esta información, recoge la fase a la que pertenece cada iteración y la fase e iteración actuales en las que se encuentra el proyecto.

Una vez obtenidos todos los datos, el fichero imprime una tabla con las iteraciones creadas. Pero no muestra todas ellas, sino las iteraciones pertenecientes a la fase seleccionada. Los selectores de fase se muestran en la parte superior de la tabla. El fichero cuenta con una función Javascript que recoge el evento de cambio de fase (clic en el [radioButton](#)), de tal forma que se muestren las iteraciones pertenecientes a la fase seleccionada.

De hecho, el fichero muestra por defecto las iteraciones pertenecientes a la fase actual en la que se encuentre el proyecto.



Nº	Nombre	Duración (en semanas)	Estado	Productos
1	Funcionalidades básicas	3	Pendiente	Listado
2	Funcionalidades adicionales	3	Pendiente	Listado
3	Gráficas y documentos	3	Pendiente	Listado
4	Funcionalidades extra	2	Pendiente	Listado

Ilustración 22. Listado de iteraciones.

Por cada iteración, se muestra la duración de la misma, junto con el estado actual en la que se encuentra. Además se muestra un desplegable con los productos que se esperan obtener al final de la iteración correspondiente. Junto al desplegable, se muestran las diferentes opciones que el usuario puede tomar: editar la iteración seleccionada, acceder a su historial o eliminarla (siempre que cuente con los permisos necesarios).

Finalmente, hay que indicar que en la parte superior de la tabla de iteraciones, se muestra la opción “Nueva iteración” que enlaza con el fichero [crear_iteracion.php](#).

4.7.2 crear_iteracion.php

Este fichero muestra un formulario con todos los datos con los que se creará la nueva iteración.

En primer lugar, el fichero accede a la base de datos para consultar la fase actual del proyecto sobre el que el usuario está trabajando, para mostrar dicha fase por defecto en el selector de fases. De igual modo, en dicho selector se mostrarán únicamente las fases igual o superiores a la fase actual.

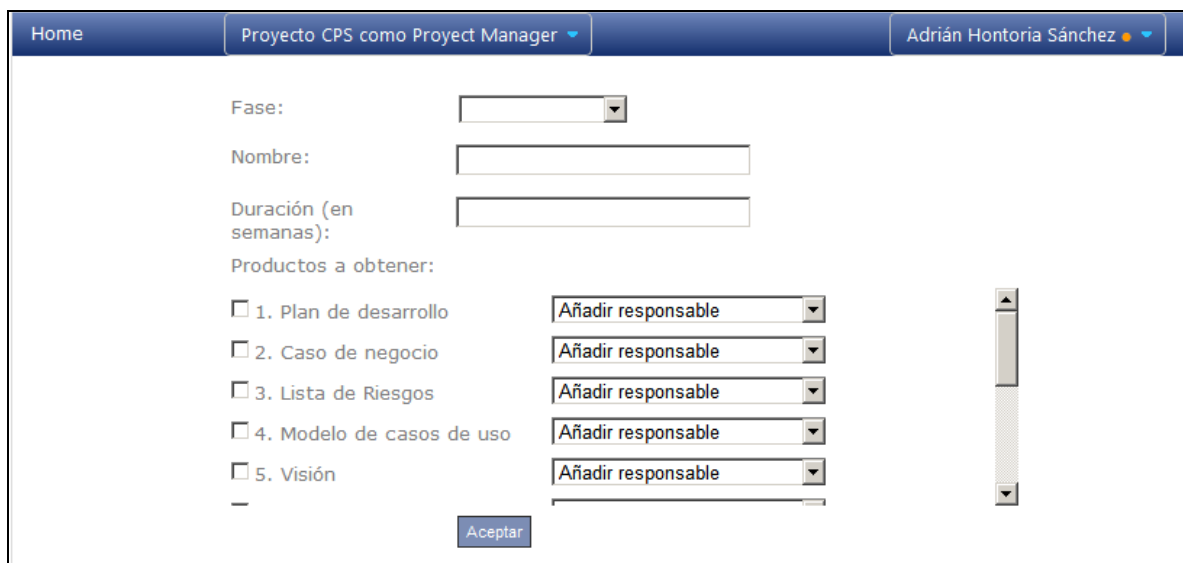


Ilustración 23. Crear iteración.

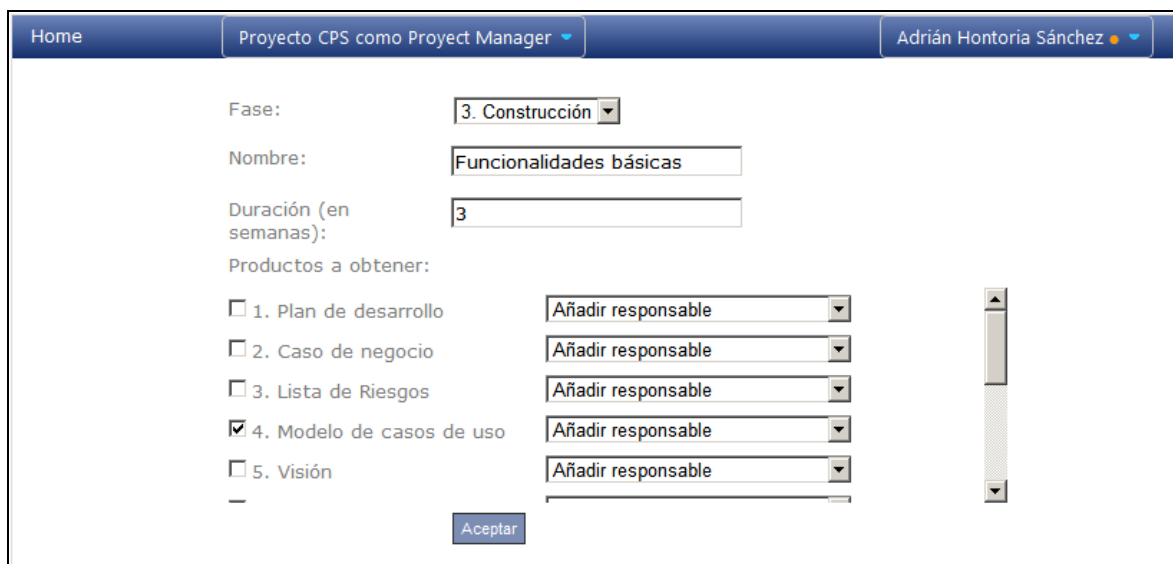
A continuación, se imprimen los campos de texto para el nombre de la iteración y la duración de la misma. Dichos campos son obligatorios, por lo que el fichero alertará al usuario si no los ha cumplimentado correctamente. Tras estos campos, se incluye el listado de productos disponibles, tras su consulta a la base de datos, para que el usuario pueda marcar los productos que se obtendrán al finalizar la iteración.

El fichero muestra también un selector en la parte derecha de cada producto, de tal forma que el usuario pueda indicar un responsable para cada producto seleccionado.

La confirmación del usuario conlleva a la creación de la nueva iteración. Esta acción es realizada por el módulo “[Gestión iteraciones](#)”.

4.7.3 editar_iteracion.php

El fichero `editar_iteracion.php` es muy similar al anterior, con la diferencia en que todos los campos aparecen cumplimentados con la información que se indicó en el momento de la creación de la iteración.



Home Proyecto CPS como Project Manager Adrián Hontoria Sánchez

Fase: 3. Construcción

Nombre: Funcionalidades básicas

Duración (en semanas): 3

Productos a obtener:

<input type="checkbox"/> 1. Plan de desarrollo	Añadir responsable
<input type="checkbox"/> 2. Caso de negocio	Añadir responsable
<input type="checkbox"/> 3. Lista de Riesgos	Añadir responsable
<input checked="" type="checkbox"/> 4. Modelo de casos de uso	Añadir responsable
<input type="checkbox"/> 5. Visión	Añadir responsable

Aceptar

Ilustración 24. Editar iteración.

Al igual que el fichero [crear_iteracion.php](#), este fichero controla que los campos obligatorios están correctamente cumplimentados, mostrando un aviso de error en caso contrario.

De igual forma, todo el proceso de edición de la iteración tras confirmar el formulario es realizado en el módulo “[Gestión iteraciones](#)”.

4.7.4 historial_iteraciones.php

Como su nombre indica, este fichero muestra el historial completo de la iteración seleccionada. En primer lugar, obtiene el identificador de la iteración a través del parámetro recibido por POST.

A continuación, el fichero realiza una consulta múltiple para acceder a todas las acciones realizadas sobre la iteración.

Home

Proyecto CPS como Project Manager

Adrián Hontoria Sánchez

Historial de la Iteración 2-Funcionalidades adicionales:

Realizado por:	Acción realizada	Acción aplicada a	Valor anterior	Valor nuevo	Fecha
Sr. Administrador Apellido	Crear Iteración	Fase 3		Iteración 2	2012-03-20 12:50:45
Sr. Administrador Apellido	Asignar	Producto Modelo de casos de uso			2012-03-20 12:50:45
Sr. Administrador Apellido	Asignar	Producto Requisitos suplementarios			2012-03-20 12:50:45

Anterior

Siguiente

Volver

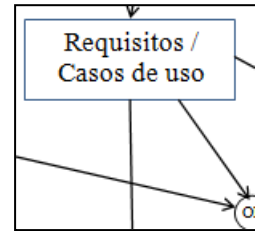
Ilustración 25. Historial de la iteración.

El fichero muestra una tabla con todos los datos obtenidos de la base de datos. Cada acción se muestra en cada fila de la tabla. El número de filas está limitado a 15 por página. Tras el listado de acciones, el fichero muestra los enlaces “Anterior” y “Siguiente” para navegar por las diferentes páginas del historial.

De forma adicional, los registros de la tabla se pueden ordenar por columna y en orden ascendente o descendente, dependiendo de la selección del icono correspondiente.

4.8. Módulo Requisitos/Casos de uso

A continuación se especifican los ficheros que componen el módulo “Requisitos/Casos de uso”. Por cada fichero se explicará la funcionalidad desarrollada así como la utilización de dicha funcionalidad por parte del usuario final.



4.8.1 requisitos.php

La función principal de este fichero es mostrar los casos de uso y requisitos creados para el proyecto seleccionado.

En primer lugar, el fichero accede al identificador del proyecto seleccionado, que se encuentra almacenado en la variable de sesión correspondiente. Gracias al identificador, el fichero accede a todos los requisitos y casos de uso creados en el proyecto.

Una vez obtenidos todos los datos, el fichero imprime dos tablas: una con los casos de uso y otra con los requisitos. Las dos tablas no se muestran de forma simultánea, sino que se ha implantado un control mediante “[radioButton](#)”, desde el que el usuario puede elegir el listado de requisitos o de casos de uso.

Home

Proyecto CPS como Project Manager

Adrián Hontoria Sánchez



Nuevo requisito

☐ Caso de uso

☒ Requisito suplementario

ID	Título	Descripción	Estado	Tipo	Subtipo	Impacto	Esfuerzo	Versión	Trazas	
R1	Gestionar proyectos	El sistema debe permitir al usuario crear, editar y eliminar proyectos.	✓	Funcionalidad	Básica	Muy alto	Medio	2	Listado	  
R2	Gestionar iteraciones	El sistema debe permitir al usuario crear, editar y eliminar iteraciones clasificadas por fases.	🕒	Funcionalidad	Básica	Alto	Medio	10	Listado	  
R3	Gestionar requisitos	El sistema debe permitir al usuario crear, editar y eliminar requisitos.	👤	Funcionalidad	Básica	Muy alto	Medio	1	Listado	  
R4	Gestionar casos de uso	El sistema debe permitir al usuario crear, editar y eliminar casos de uso.	✗	Funcionalidad	Básica	Muy alto	Medio	1	Listado	  
R5	Gestionar trazabilidad	El sistema permitirá al usuario establecer relaciones de dependencia entre requisitos y casos de uso a través de una matriz de trazabilidad.	?	Funcionalidad	Básica	Alto	Alto	1	Listado	  
R6	Gestionar entregables	El sistema debe permitir al usuario crear, editar y eliminar entregables.	✓	Funcionalidad	Básica	Muy alto	Alto	1	Listado	  
R7	Datos casos de uso	El sistema obligará al usuario a indicar un título y una imagen asociada al caso de uso que se creará o editará. En caso de no indicar alguno de estos campos, el sistema avisará al usuario a través de una alerta visual.	✓	Funcionalidad	Adicional	Medio	Medio	2	Listado	  
R8	Generación de entregables	El sistema permitirá generar de dos formas los entregables: de forma manual y de forma automática.	✓	Funcionalidad	Gráficas y documentos	Medio	Alto	2	Listado	  

Ilustración 26. Listado de requisitos.

Home

Proyecto CPS como Project Manager

Adrián Hontoria Sánchez



[Nuevo requisito](#)

☒ Caso de uso

☐ Requisito suplementario

ID	Título	Estado	Versión	
CU1	Gestionar proyectos	✓	1	  
CU2	Gestionar iteraciones	✓	1	  
CU3	Gestionar requisitos	✓	1	  
CU4	Gestionar casos de uso	✓	2	  
CU5	Gestionar trazabilidad	✓	1	  
CU6	Gestionar entregables	✓	1	  
CU7	Personalizar gráficas	✓	1	  
CU8	Gestionar avisos	✓	1	  

Anterior

Siguiente

Volver

Ilustración 27. Listado de casos de uso.

Finalmente, hay que indicar que en la parte superior de la tabla de requisitos o de casos de uso, se muestra la opción “Nuevo requisito” que enlaza con el fichero [crear_requisito.php](#).

4.8.2 crear_requisito.php

Este fichero es común para la creación de requisitos y de casos de uso. Lo primero que muestra el fichero es un selector del elemento a crear: caso de uso o requisito. Dependiendo de la opción seleccionada, se muestra el formulario de nuevo requisito o el formulario de nuevo caso de uso.

El formulario de caso de uso es muy sencillo. Consta de dos campos: título del nuevo caso de uso e imagen del diagrama del caso de uso. Ambos campos son obligatorios por lo que el fichero alertará si no se han cumplimentado correctamente.

El campo de la imagen consiste en un campo de formulario de tipo “file”, en el que el usuario podrá seleccionar cualquier archivo local de su ordenador para cargarlo a la base de datos.

Por otro lado, el formulario de creación de un requisito muestra otro conjunto de campos y listados necesarios para el nuevo requisito.



Ilustración 28. Crear caso de uso.

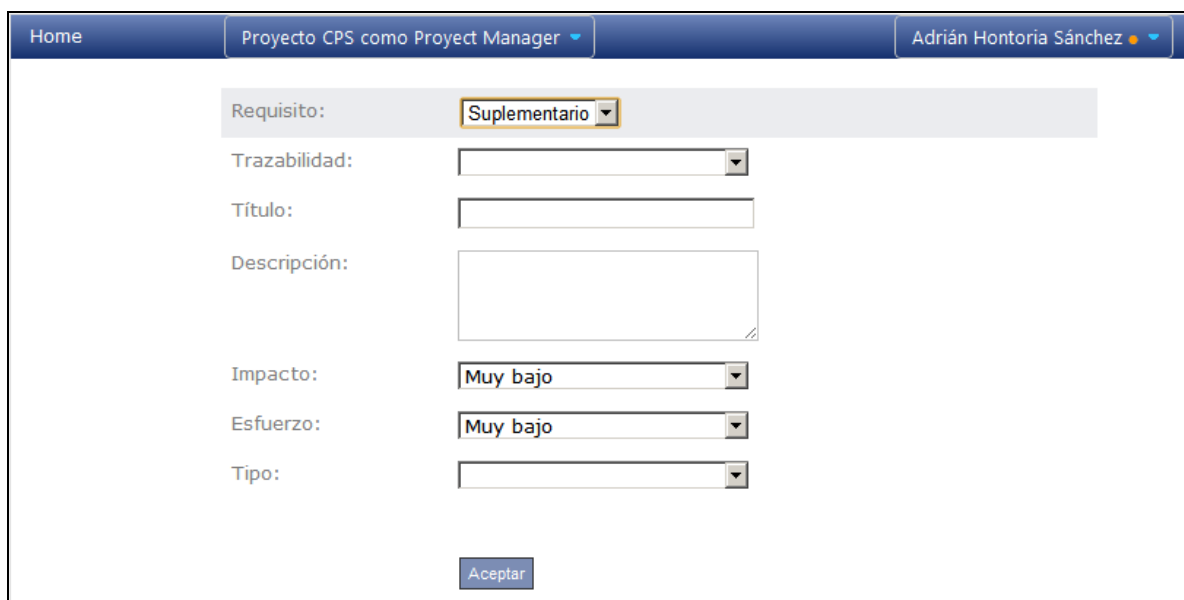


Ilustración 29. Crear requisito.

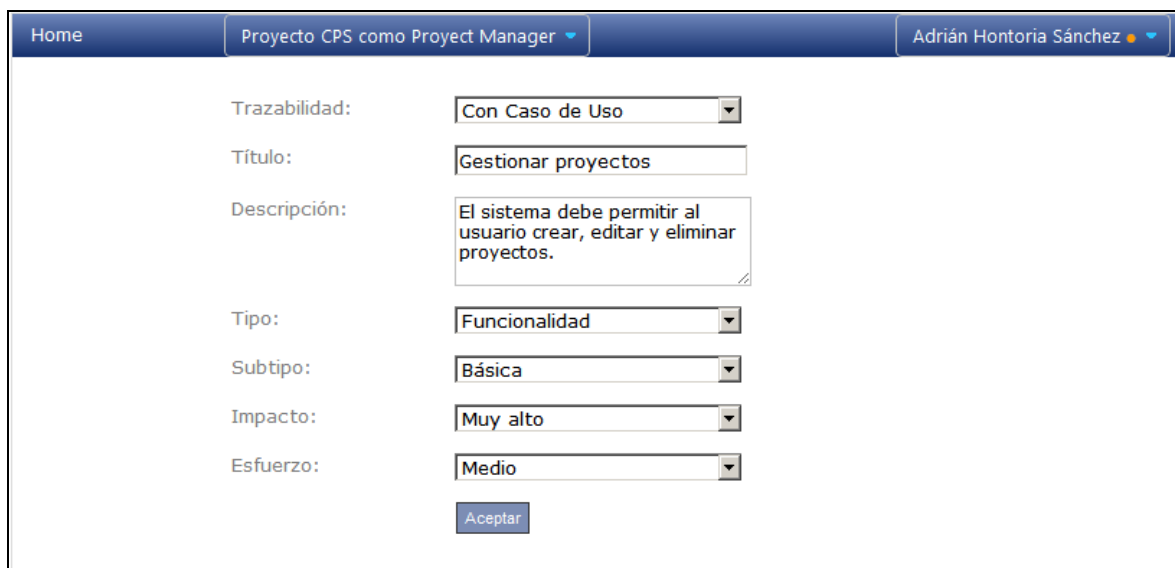
Todos los campos son obligatorios, por lo que el fichero alertará al usuario si no los ha cumplimentado correctamente.

La confirmación del usuario conlleva a la creación del nuevo caso de uso o requisito. Esta acción es realizada por el módulo “[Gestión requisitos](#)”.

4.8.3 editar_requisito.php

El fichero editar_requisito.php es muy similar al anterior, con dos diferencias:

- En este caso, el fichero editar_requisito.php tiene la función de editar únicamente requisitos y no casos de uso.
- Todos los campos aparecen cumplimentados con la información que se indicó en el momento de la creación del requisito.



Home	Proyecto CPS como Project Manager	Adrián Hontoria Sánchez
Trazabilidad:	Con Caso de Uso	
Título:	Gestionar proyectos	
Descripción:	El sistema debe permitir al usuario crear, editar y eliminar proyectos.	
Tipo:	Funcionalidad	
Subtipo:	Básica	
Impacto:	Muy alto	
Esfuerzo:	Medio	
<input type="button" value="Aceptar"/>		

Ilustración 30. Editar requisito.

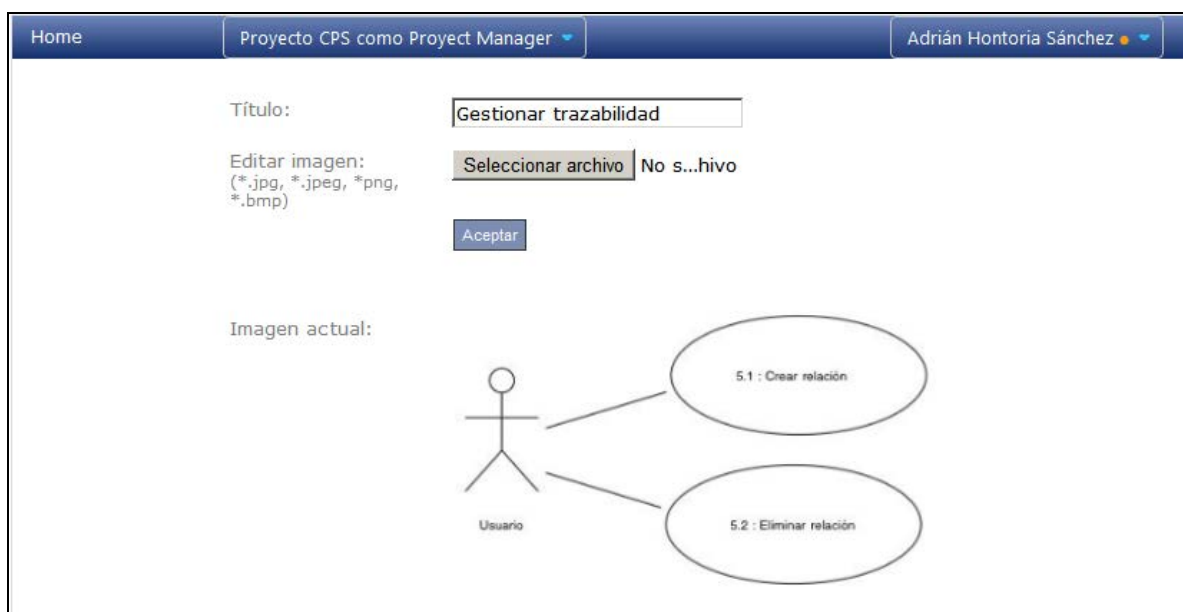
Al igual que el fichero [crear_requisito.php](#), este fichero controla que los campos obligatorios están correctamente cumplimentados, mostrando un aviso de error en caso contrario.

De igual forma, todo el proceso de edición del requisito, tras confirmar el formulario, es realizado en el módulo “[Gestión requisitos](#)”.

4.8.4 editar_caso.php

El fichero `editar_caso.php` es muy similar al fichero [crear_requisito.php](#), con tres diferencias:

- En este caso, el fichero `editar_caso.php` tiene la función de editar únicamente casos de uso y no requisitos.
- Todos los campos aparecen cumplimentados con la información que se indicó en el momento de la creación del caso de uso.
- Se añade una imagen previa del diagrama indicado en el momento de la creación del requisito, con un enlace a una nueva ventana que muestra el diagrama a tamaño completo.



Home Proyecto CPS como Project Manager Adrián Hontoria Sánchez

Título: Gestionar trazabilidad

Editar imagen: (*.jpg, *.jpeg, *.png, *.bmp) Seleccionar archivo No s...hivo

Aceptar

Imagen actual:

UML Use Case Diagram:

```
graph LR
    Usuario((Usuario)) --- UC1(5.1 : Crear relación)
    Usuario --- UC2(5.2 : Eliminar relación)
```

Ilustración 31. Editar caso de uso.

Al igual que el fichero [crear_requisito.php](#), este fichero controla que los campos obligatorios están correctamente cumplimentados, mostrando un aviso de error en caso contrario.

De igual forma, todo el proceso de edición del caso de uso, tras confirmar el formulario, es realizado en el módulo “[Gestión requisitos](#)”.

4.8.5 historial_requisitos.php

Como su nombre indica, este fichero muestra el historial completo del requisito o caso de uso seleccionado. En primer lugar, obtiene el identificador del requisito o caso de uso a través del parámetro recibido por POST. De igual forma, obtiene un control que estuvo oculto en el fichero [requisitos.php](#) indicando si el identificador recibido se corresponde con un requisito o con un caso de uso.

A continuación, el fichero realiza una consulta múltiple para acceder a todas las acciones realizadas sobre el requisito o caso de uso.

Home

Proyecto CPS como Project Manager

Adrián Hontoria Sánchez

Historial del Requisito R2-Gestionar iteraciones:

Realizado por: ▼▲	Acción realizada ▼▲	Acción aplicada a ▼▲	Valor anterior ▼▲	Valor nuevo ▼▲	Fecha ▼▲
Luis Miguel Sánchez García	Aceptar	Requisito Gestionar iteraciones			2012-03-20 14:54:52
Adrián Hontoria Sánchez	Publicar a cliente	Requisito Gestionar iteraciones			2012-03-20 14:51:47
Adrián Hontoria Sánchez	Editar	Impacto	5	4	2012-03-20 13:49:10
Adrián Hontoria Sánchez	Editar	Título	Gestionar iteracione	Gestionar iteraciones	2012-03-20 13:47:27
Adrián Hontoria Sánchez	Editar	Subtipo	Adicional	Básica	2012-03-20 13:40:21
Adrián Hontoria Sánchez	Editar	Subtipo		Adicional	2012-03-20 13:40:04
Adrián Hontoria Sánchez	Editar	Subtipo	Básica		2012-03-20 13:39:47
Adrián Hontoria Sánchez	Editar	Título	Gestionar iteracione	Gestionar iteraciones	2012-03-20 13:39:23
Adrián Hontoria Sánchez	Editar	Título	Gestionar iteracione	Gestionar iteraciones	2012-03-20 13:28:40
Adrián Hontoria Sánchez	Editar	Subtipo	Básica	27	2012-03-20 13:12:27
Adrián Hontoria Sánchez	Editar	Título	Gestión de iteracion	Gestionar iteraciones	2012-03-20 13:12:27
Adrián Hontoria Sánchez	Alta	Funcionalidad - Básica			2012-03-20 13:01:46

Anterior

Siguiente

Volver

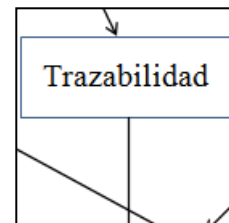
Ilustración 32. Historial de requisitos.

El fichero muestra una tabla con todos los datos obtenidos de la base de datos. Cada acción se muestra en cada fila de la tabla. El número de filas está limitado a 15 por página. Tras el listado de acciones, el fichero muestra los enlaces “Anterior” y “Siguiente” para navegar por las diferentes páginas del historial.

De forma adicional, los registros de la tabla se pueden ordenar por columna y en orden ascendente o descendente, dependiendo de la selección del icono correspondiente.

4.9. Módulo Trazabilidad

A continuación se especifican los ficheros que componen el módulo “Trazabilidad”. Por cada fichero se explicará la funcionalidad desarrollada así como la utilización de dicha funcionalidad por parte del usuario final.



4.9.1 trazabilidad.php

La función principal de este fichero es mostrar la matriz de trazabilidad con las dependencias entre los requisitos y los casos de uso.

En primer lugar, el fichero accede al identificador del proyecto actual almacenado en la variable de sesión correspondiente. A continuación accede a todos los casos de uso y a todos los requisitos del proyecto. Hay que indicar que los requisitos consultados son únicamente los que cuentan con la propiedad trazabilidad con casos de uso.

Tras obtener los casos de uso y los requisitos, se realiza una nueva consulta recorriendo cada requisito para obtener la dependencia con los casos de uso. Por cada requisito comprobado, se imprime una fila de la tabla que consistirá en la matriz de trazabilidad.

Home		Proyecto CPS como Project Manager ▾				Adrián Hontoria Sánchez ● ▾			
		CU8: gestionar_avisos	CU7: personalizar_grafico	CU6: gestionar_entregables	CU5: gestionar_trazabilidad	CU4: gestionar_casos_de_uso	CU3: gestionar_requisitos	CU2: gestionar_iteraciones	CU1: gestionar_proyectos
R1: Gestionar proyectos									✓
R2: Gestionar iteraciones								✓	
R3: Gestionar requisitos							✓		
R4: Gestionar casos de uso						✓			
R5: Gestionar trazabilidad					✓				
R6: Gestionar entregables			✓						
R7: Datos casos de uso						✓			
R8: Generación de entregables			✓						
R9: Generación manual de entregables			✓						
R10: Generación automática de entregables			✓						

Ilustración 33. Matriz de trazabilidad.

En la parte inferior de la matriz, se muestra el enlace al fichero [editar_trazabilidad.php](#) para modificar dicha matriz.

4.9.2 editar_trazabilidad.php

Este fichero es muy similar al anterior con una única diferencia: en lugar de mostrar los iconos que afirman la dependencia de un requisito con un caso de uso, se muestra un control [checkBox](#) por cada relación requisito-caso de uso.

Home		Proyecto CPS como Project Manager ▾		Adrián Hontoria Sánchez ● ▾					
		CU8: gestionar_avisos	CU7: personalizar_grafica	CU6: gestionar_entregables	CU5: gestionar_trazabilidad	CU4: gestionar_casos_de_uso	CU3: gestionar_requisitos	CU2: gestionar_iteraciones	CU1: gestionar_proyectos
R1: Gestionar proyectos		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
R2: Gestionar iteraciones		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
R3: Gestionar requisitos		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
R4: Gestionar casos de uso		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
R5: Gestionar trazabilidad		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
R6: Gestionar entregables		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
R7: Datos casos de uso		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
R8: Generación de entregables		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
R9: Generación manual de entregables		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
R10: Generación automática de entregables		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

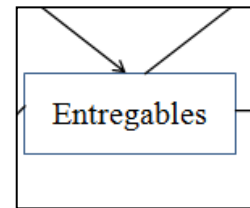
Ilustración 34. Edición de la matriz de trazabilidad.

Gracias a estos controles, el usuario podrá marcar o desmarcar las dependencias entre los requisitos y casos de uso.

A la hora de confirmar los cambios, todo el proceso es realizado por el módulo “[Gestión trazabilidad](#)”.

4.10. Módulo Entregables

A continuación se especifican los ficheros que componen el módulo “Entregables”. Por cada fichero se explicará la funcionalidad desarrollada así como la utilización de dicha funcionalidad por parte del usuario final.



4.10.1 entregables.php

La función principal de este fichero es mostrar los entregables creados para el proyecto seleccionado.

En primer lugar, el fichero accede al identificador del proyecto seleccionado, que se encuentra almacenado en la variable de sesión correspondiente. Gracias al identificador, el fichero accede a todos los entregables creados en el proyecto.

Una vez obtenidos todos los datos, el fichero imprime una tabla con los entregables creados, mostrando todas sus propiedades junto con las acciones a realizar para cada entregable.

Home

Proyecto CPS como Project Manager

Adrián Hontoria Sánchez

Nuevo entregable

ID	Título	Descripción	Estado	Fase	Iteración	Versión	
E2	Modelo de casos de uso			Construcción	1	3	
E1	Requisitos suplementarios			Construcción	1	7	

Anterior

Siguiente

Volver

Ilustración 35. Listado de entregables.

Finalmente, hay que indicar que en la parte superior de la tabla de entregables, se muestra la opción “Nuevo entregable” que enlaza con el fichero [crear_entregable.php](#).

4.10.2 crear_entregable.php

Este fichero muestra un formulario muy simple para crear un nuevo entregable. En primer lugar, el fichero accede a la base de datos para comprobar qué entregables se pueden crear dependiendo de la fase y la iteración en la que se encuentre el proyecto. Estos entregables se mostrarán en un selector.

A la hora de seleccionar uno de los entregables de la lista, el fichero comprueba de qué tipo de entregable se trata, ya que puede tratarse de un entregable que se genere de forma automática, o de lo contrario, que se genere de forma manual. Tras la comprobación, el fichero avisa al usuario de su deber de indicar al menos un documento externo si se trata de un entregable “manual”.

De forma adicional, la última opción del listado es “Otro”. Al seleccionar dicha opción, una función Javascript muestra dos campos obligatorios para la creación del nuevo entregable no registrado en el proyecto: título y descripción.

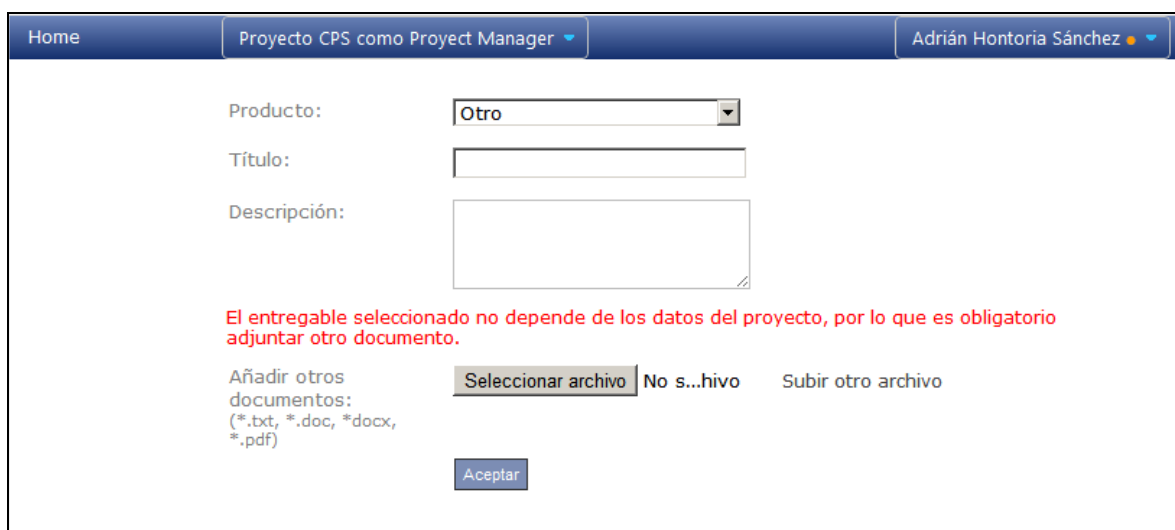


Ilustración 36. Crear otro entregable.

Por último, se indica el campo “file” desde el que el usuario puede cargar cualquier documento local.

La confirmación del usuario conlleva a la creación del nuevo entregable. Esta acción es realizada por el módulo “[Gestión entregables](#)”.

4.10.3 editar_entregable.php

El fichero editar_entregable.php es muy similar al anterior, cuya diferencia reside en que en esta ocasión no es necesario indicar el producto a generar, ya que se trata del entregable a actualizar.

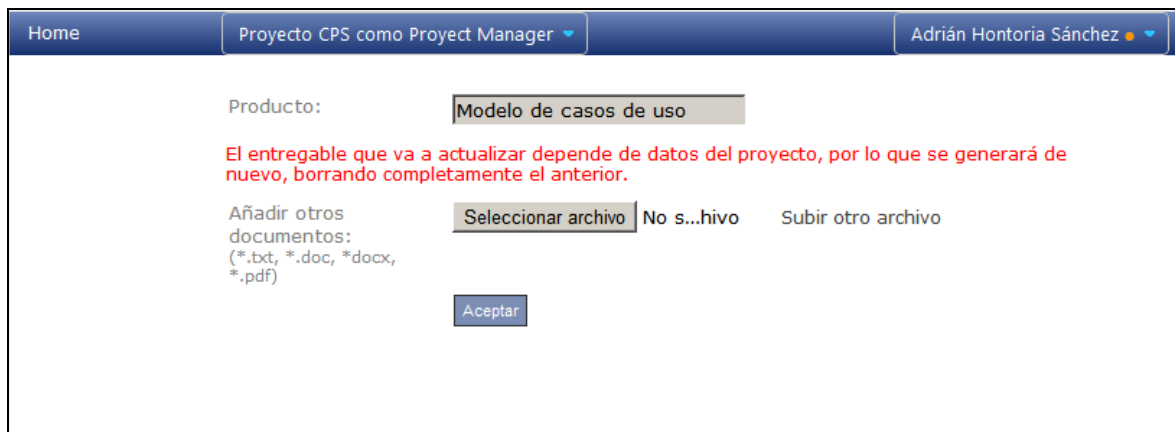


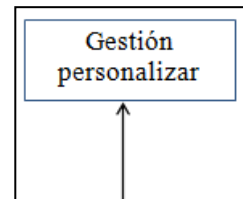
Ilustración 37. Actualizar entregable.

En este caso, el fichero controla si el entregable a actualizar se generó por primera vez con datos internos de la aplicación o con datos externos. En el primer caso, el fichero muestra un aviso indicando que el entregable se actualizará con los nuevos datos del proyecto, borrando los datos anteriores. En el segundo caso, los nuevos documentos indicados se añadirán al entregable anterior.

De igual forma, todo el proceso de actualización del entregable tras confirmar el formulario es realizado en el módulo “[Gestión entregables](#)”.

4.11. Módulo Gestión Personalizar

A continuación se especifican los ficheros que componen el módulo “Gestión Personalizar”. Por cada fichero se explicará la funcionalidad desarrollada así como la utilización de dicha funcionalidad por parte del usuario final.



4.11.1 `gestion_personalizar.php`

Este fichero se encarga de llevar a cabo la operación de activación o desactivación de gráficas para un proyecto y usuario concretos.

El fichero es llamado desde [personalizar.php](#) y recibe los siguientes parámetros:

- Identificador del proyecto al que activar o desactivar las gráficas.
- El [array](#) de gráficas que se desean activar.

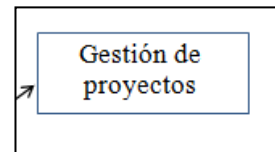
En primer lugar, se recorre elemento por elemento el array recibido. Por cada gráfica se realiza una llamada a la función “`activarGrafica(id_usuario, id_proyecto, id_grafica)`”. Así, tras recorrer todo el array, las gráficas indicadas ya han sido activadas.

En segundo lugar, hay que desactivar las gráficas que no se hayan indicado. Para ello, se hace una consulta para obtener un listado con todas las gráficas activadas. Si una gráfica de la lista no existe en el array recibido por parámetro, significa que hay que eliminarla de la base de datos (relación usuario-proyecto-gráfica). Para ello, se realiza una llamada a la función “`desactivarGrafica(id_usuario, id_proyecto, id_grafica)`”.

Finalmente, se envía al usuario al fichero [personalizar.php](#) indicando el resultado de la operación a través de un parámetro GET.

4.12. Módulo Gestión Proyectos

A continuación se especifican los ficheros que componen el módulo “Gestión Proyectos”. Por cada fichero se explicará la funcionalidad desarrollada así como la utilización de dicha funcionalidad por parte del usuario final.



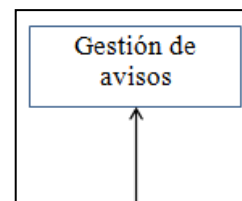
4.12.1 `gestion_proyectos.php`

Este fichero se encarga de llevar a cabo las siguientes acciones. Cada fichero de origen dispone de un control oculto para que el fichero `gestion_proyectos.php` pueda seleccionar la acción adecuada:

- **Crear un nuevo proyecto:** para ello, recibe por parámetro POST el título y la descripción del nuevo proyecto y se realiza una llamada a la función “`crear_proyecto(titulo, descripción)`”. El resultado se devuelve al fichero [iteraciones.php](#), ya que la creación del Plan de Iteraciones es el siguiente paso tras la creación de un proyecto.
- **Editar un proyecto existente:** recibe por parámetro el nuevo título, la nueva descripción y el identificador del proyecto existente. A continuación se realiza una llamada a la función “`editar_proyecto(id_proyecto, titulo, descripción)`”. El resultado se devuelve al fichero [proyectos.php](#).
- **Asignar participantes a un proyecto existente:** recibe por parámetro el identificador del proyecto, el [array](#) con los nuevos usuarios para añadir al proyecto y los roles de cada uno de los usuarios. El fichero recorre el array elemento a elemento y realiza la llamada a la función “`asignar_proyecto(id_usuario_identificado, id_proyecto, id_usuario_array, accion, fecha, id_rol)`” por cada usuario indicando la acción de “Alta”. El resultado se devuelve al fichero [proyectos.php](#).
- **Desasignar participantes de un proyecto:** utilizando el array descrito en el punto anterior, el fichero compara la lista de usuarios participantes en la base de datos con la del array. Si un usuario existe en la base de datos pero no en el array indicado, significa que hay que desasignarle del proyecto indicado. Para ello, se realiza la llamada a la función “`asignar_proyecto(id_usuario_identificado, id_proyecto, id_usuario_array, accion, fecha, id_rol)`” indicando la acción de “Baja”. El resultado se devuelve al fichero [proyectos.php](#).

4.13. Módulo Gestión Avisos

A continuación se especifican los ficheros que componen el módulo “Gestión Avisos”. Por cada fichero se explicará la funcionalidad desarrollada así como la utilización de dicha funcionalidad por parte del usuario final.



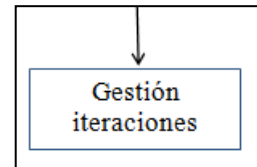
4.13.1 `gestion_avisos.php`

Este fichero se encarga de llevar a cabo las acciones de crear un nuevo aviso y eliminar un aviso existente:

- **Crear un nuevo aviso:** recibe por parámetro el identificador del emisor, el identificador del receptor, el mensaje y el identificador del proyecto. Con estos datos se realiza la llamada a la función “`crearAviso(id_proyecto, id_emisor, id_receptor, mensaje)`”. El resultado obtenido se devuelve al fichero [avisos.php](#).
- **Eliminar aviso existente:** recibe por parámetro el identificador del aviso a eliminar y realiza la llamada a la función “`eliminarAviso(id_aviso)`”. El resultado del proceso se devuelve al fichero [avisos.php](#).

4.14. Módulo Gestión Iteraciones

A continuación se especifican los ficheros que componen el módulo “Gestión Iteraciones”. Por cada fichero se explicará la funcionalidad desarrollada así como la utilización de dicha funcionalidad por parte del usuario final.



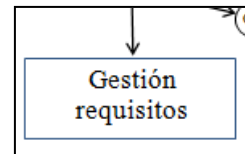
4.14.1 `gestion_iteraciones.php`

Este fichero se encarga de llevar a cabo las siguientes acciones. Cada fichero de origen dispone de un control oculto para que el fichero `gestion_iteraciones.php` pueda seleccionar la acción adecuada:

- **Crear una nueva iteración:** recibe por parámetro el nombre, la duración, la fase y el [array](#) de productos de la iteración a crear.
 - En primer lugar se crea la iteración con la llamada “`crear_iteracion(id_proyecto, id_fase, nombre, duracion, id_usuario_identificado)`”.
 - A continuación, por cada producto indicado en el array recibido, se asigna dicho producto a la iteración creada mediante la función “`asignar_iteracion_producto(id_proyecto, id_fase, id_iteracion, id_producto, id_usuario_identificado, id_responsable_producto)`”.
 - El resultado de ambas funciones se devuelve al fichero [iteraciones.php](#).
- **Editar una iteración existente:** recibe por parámetro el nombre, la duración, la fase y el array de productos de la iteración a modificar.
 - Con los datos anteriores, se realiza la llamada a la función “`editar_iteracion(id_proyecto, id_fase, id_iteracion, nombre, duracion, id_usuario_identificado)`”.
 - A continuación, por cada producto indicado en el array recibido, se asigna dicho producto a la iteración indicada mediante la función “`asignar_iteracion_producto(id_proyecto, id_fase, id_iteracion, id_producto, id_usuario_identificado, id_responsable_producto)`”.
 - Por último, se compara el listado de productos de la iteración en la base de datos y se comparan con el array recibido. Si un producto existente en la base de datos no está indicado en el array recibido, significa que hay que borrarlo de la iteración. Para ello, se llama a la función “`desasignar_iteracion_producto(id_proyecto, id_fase, id_iteracion, id_producto, id_usuario_identificado)`”.
 - El resultado de las tres funciones se devuelve al fichero [iteraciones.php](#).

4.15. Módulo Gestión Requisitos

A continuación se especifican los ficheros que componen el módulo “Gestión Requisitos”. Por cada fichero se explicará la funcionalidad desarrollada así como la utilización de dicha funcionalidad por parte del usuario final.



4.15.1 `gestion_requisitos.php`

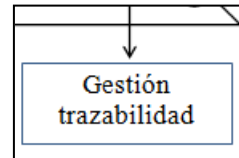
Este fichero se encarga de llevar a cabo las siguientes acciones. Cada fichero de origen dispone de un control oculto para que el fichero `gestion_requisitos.php` pueda seleccionar la acción adecuada:

- **Crear un nuevo requisito:** recibe por parámetro el título, la descripción, el tipo, el subtipo, el impacto, el esfuerzo y la trazabilidad del nuevo requisito. Con todos los datos, se crea el requisito gracias a la función “`crear_requisito(titulo, descripcion, tipo, subtipo, impacto, esfuerzo, id_proyecto, id_usuario_identificado, trazabilidad)`”. El resultado es enviado al fichero [requisitos.php](#).
- **Editar un requisito existente:** recibe por parámetro el título, la descripción, el tipo, el subtipo, el impacto, el esfuerzo y la trazabilidad del requisito a modificar. Con todos los datos, se edita el requisito gracias a la función “`editar_requisito(id_requisito, titulo, descripcion, tipo, subtipo, impacto, esfuerzo, id_proyecto, id_usuario_identificado, trazabilidad)`”. El resultado es enviado al fichero [requisitos.php](#).
- **Crear un nuevo caso de uso:** recibe por parámetro el título y la imagen asociada al nuevo caso de uso.
 - En primer lugar, se verifica que el archivo que contiene la imagen ha sido cargado correctamente mediante la función “`is_uploaded_file(archivo)`”.
 - En segundo lugar, se mueve el archivo cargado, desde la carpeta temporal del servidor a la carpeta “`./img/casos_uso/`” mediante la función “`move_uploaded_file(archivo, nuevo_directorio)`”.
 - Por último, se crea el caso de uso en la base de datos mediante la función “`crear_caso(titulo, ruta_imagen, id_proyecto, id_usuario_identificado)`”.
 - El resultado es enviado al fichero [requisitos.php](#).
- **Editar un caso de uso existente:** recibe por parámetro el título y la imagen que quedará asociada al caso de uso que se desea editar.
 - En primer lugar, se verifica que el archivo que contiene la imagen ha sido cargado correctamente mediante la función “`is_uploaded_file(archivo)`”.
 - En segundo lugar, se mueve el archivo cargado, desde la carpeta temporal del servidor a la carpeta “`./img/casos_uso/`” mediante la función “`move_uploaded_file(archivo, nuevo_directorio)`”.

- Por último, se edita el caso de uso en la base de datos mediante la función “editar_caso(id_caso, titulo, ruta_imagen, id_proyecto, id_usuario_identificado)”.
- El resultado es enviado al fichero [requisitos.php](#).

4.16. Módulo Gestión Trazabilidad

A continuación se especifican los ficheros que componen el módulo “Gestión Trazabilidad”. Por cada fichero se explicará la funcionalidad desarrollada así como la utilización de dicha funcionalidad por parte del usuario final.



4.16.1 gestion_trazabilidad.php

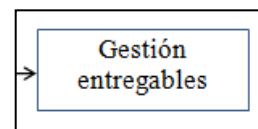
Este fichero se encarga de activar o desactivar las dependencias entre requisitos y casos de uso. Para ello, recibe por parámetro el [array](#) de las relaciones requisito-caso de uso activadas.

El fichero recorre elemento a elemento dicho array y crea la nueva dependencia gracias a la función “trazar(id_req, id_caso)”.

Tras la creación de las trazas, el fichero procede a la eliminación de las trazas que el usuario ha indicado. Para ello, el fichero obtiene la lista de trazas existentes en la base de datos y elimina todas aquellas que no se encuentren en el array recibido mediante la función “eliminar_traza(id_requisito, id_caso)”. Finalmente, el resultado de ambas funciones se devuelve al fichero [trazabilidad.php](#).

4.17. Módulo Gestión Entregables

A continuación se especifican los ficheros que componen el módulo “Gestión Entregables”. Por cada fichero se explicará la funcionalidad desarrollada así como la utilización de dicha funcionalidad por parte del usuario final.



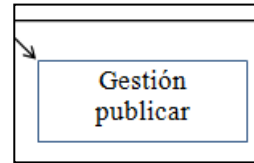
4.17.1 `gestion_entregables.php`

Este fichero se encarga de llevar a cabo las siguientes acciones. Cada fichero de origen dispone de un control oculto para que el fichero `gestion_entregables.php` pueda seleccionar la acción adecuada:

- **Crear un nuevo entregable:** recibe por parámetro el identificador del entregable a generar y un [array](#) con los documentos opcionales que el usuario quiera adjuntar. A continuación realiza una de las siguientes acciones:
 - Si el entregable a generar es otro producto diferente a los listados en el sistema, el fichero obtiene el título y la descripción por parámetro.
 - Si el entregable a generar es un producto por defecto en el sistema, el fichero obtiene el título y la descripción a partir de una consulta a la base de datos.
 - Finalmente se crea el nuevo entregable gracias a la función “`crear_entregable(titulo, descripcion, array_documentos, num_documentos, id_proyecto, id_usuario_identificado, producto)`”.
 - El resultado de la función se devuelve al fichero [entregables.php](#).
- **Actualizar un entregable existente:** recibe por parámetro el identificador del entregable a actualizar y el array con los nuevos documentos a adjuntar. El entregable es actualizado gracias a la función “`actualizar_entregable(id_entregable, array_documentos, num_documentos, id_proyecto, id_usuario_identificado)`”. El resultado de la función se devuelve al fichero [entregables.php](#).

4.18. Módulo Gestión Publicar

A continuación se especifican los ficheros que componen el módulo “Gestión Publicar”. Por cada fichero se explicará la funcionalidad desarrollada así como la utilización de dicha funcionalidad por parte del usuario final.



4.18.1 `gestion_publicar.php`

Este fichero se encarga de las publicaciones y por tanto de los cambios de estado, de los requisitos y los casos de uso. Las diferentes acciones que realiza son las siguientes:

- **Publicar caso de uso al Project Manager:** recibe por parámetro el identificador del caso de uso a mandar a revisión.
 - Con el identificador, se obtiene el nombre del caso de uso y actualiza su estado mediante la función “`publicarCasoPM(id_caso, titulo, id_usuario, id_proyecto)`”.
 - A continuación genera el aviso automático dirigido a todos los usuarios con el rol Project Manager del proyecto actual mediante la función “`crearAviso(id_proyecto, id_emisor, id_pm, mensaje)`”.
 - Finalmente, devuelve el resultado de las funciones al fichero [requisitos.php](#).
- **Publicar requisito al Project Manager:** recibe por parámetro el identificador del requisito a mandar a revisión.
 - Con el identificador, se obtiene el nombre del requisito y actualiza su estado mediante la función “`publicarRequisitoPM(id_requisito, titulo, id_usuario, id_proyecto)`”.
 - A continuación genera el aviso automático dirigido a todos los usuarios con el rol Project Manager del proyecto actual mediante la función “`crearAviso(id_proyecto, id_emisor, id_pm, mensaje)`”.
 - Finalmente, devuelve el resultado de las funciones al fichero [requisitos.php](#).
- **Publicar caso de uso al Propietario:** recibe por parámetro el identificador del caso de uso a publicar al cliente.
 - Con el identificador, se obtiene el nombre del caso de uso y actualiza su estado mediante la función “`publicarCasoCliente(id_caso, titulo, id_usuario, id_proyecto)`”.
 - A continuación genera el aviso automático dirigido a todos los usuarios con el rol Propietario del proyecto actual mediante la función “`crearAviso(id_proyecto, id_emisor, id_propietario, mensaje)`”.

- Finalmente, devuelve el resultado de las funciones al fichero [requisitos.php](#).
- **Publicar requisito al Propietario:** recibe por parámetro el identificador del requisito a publicar al cliente.
 - Con el identificador, se obtiene el nombre del requisito y actualiza su estado mediante la función “publicarRequisitoCliente(id_requisito, titulo, id_usuario, id_proyecto)”.
 - A continuación genera el aviso automático dirigido a todos los usuarios con el rol Propietario del proyecto actual mediante la función “crearAviso(id_proyecto, id_emisor, id_propietario, mensaje)”.
 - Finalmente, devuelve el resultado de las funciones al fichero [requisitos.php](#).
- **Aceptar caso de uso:** recibe por parámetro el identificador del caso de uso a aceptar.
 - Con el identificador, se obtiene el nombre del caso de uso y actualiza su estado mediante la función “acceptarCaso(id_caso, titulo, id_usuario, id_proyecto)”.
 - A continuación genera el aviso automático dirigido a todos los usuarios con el rol Project Manager del proyecto actual mediante la función “crearAviso(id_proyecto, id_emisor, id_pm, mensaje)”.
 - Finalmente, devuelve el resultado de las funciones al fichero [requisitos.php](#).
- **Aceptar requisito:** recibe por parámetro el identificador del requisito a aceptar.
 - Con el identificador, se obtiene el nombre del requisito y actualiza su estado mediante la función “acceptarRequisito(id_requisito, titulo, id_usuario, id_proyecto)”.
 - A continuación genera el aviso automático dirigido a todos los usuarios con el rol Project Manager del proyecto actual mediante la función “crearAviso(id_proyecto, id_emisor, id_pm, mensaje)”.
 - Finalmente, devuelve el resultado de las funciones al fichero [requisitos.php](#).
- **Rechazar caso de uso:** recibe por parámetro el identificador del caso de uso a rechazar y la razón del rechazo si ha sido indicada.
 - Con el identificador, se obtiene el nombre del caso de uso y actualiza su estado mediante la función “rechazarCaso(id_caso, titulo, id_usuario, id_proyecto)”.

- A continuación genera el aviso automático, incluyendo la razón del rechazo, dirigido a todos los usuarios con el rol Project Manager del proyecto actual mediante la función “crearAviso(id_proyecto, id_emisor, id_pm, mensaje)”.
 - Finalmente, devuelve el resultado de las funciones al fichero [requisitos.php](#).
- **Rechazar requisito:** recibe por parámetro el identificador del requisito a rechazar y la razón del rechazo si ha sido indicada.
 - Con el identificador, se obtiene el nombre del requisito y actualiza su estado mediante la función “rechazarRequisito(id_requisito, titulo, id_usuario, id_proyecto)”.
 - A continuación genera el aviso automático, incluyendo la razón del rechazo, dirigido a todos los usuarios con el rol Project Manager del proyecto actual mediante la función “crearAviso(id_proyecto, id_emisor, id_pm, mensaje)”.
 - Finalmente, devuelve el resultado de las funciones al fichero [requisitos.php](#).

4.18.2 **gestion_publicar_entregable.php**

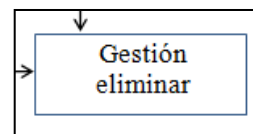
Este fichero se encarga de las publicaciones y por tanto de los cambios de estado, de los entregables. Las diferentes acciones que realiza son las siguientes:

- **Publicar entregable al Project Manager:** recibe por parámetro el identificador del entregable a mandar a revisión.
 - Con el identificador, se obtiene el nombre del entregable y se actualiza su estado mediante la función “publicarEntregablePM(id_entregable, titulo, id_usuario, id_proyecto)”.
 - A continuación genera el aviso automático dirigido a todos los usuarios con el rol Project Manager del proyecto actual mediante la función “crearAviso(id_proyecto, id_emisor, id_pm, mensaje)”.
 - Finalmente, devuelve el resultado de las funciones al fichero [entregables.php](#).
- **Publicar entregable al Propietario:** recibe por parámetro el identificador del entregable a publicar al cliente.
 - Con el identificador, se obtiene el nombre del entregable y actualiza su estado mediante la función “publicarEntregableCliente(id_entregable, titulo, id_usuario, id_proyecto)”.
 - A continuación genera el aviso automático dirigido a todos los usuarios con el rol Propietario del proyecto actual mediante la función “crearAviso(id_proyecto, id_emisor, id_propietario, mensaje)”.

- Finalmente, devuelve el resultado de las funciones al fichero [entregables.php](#).
- **Aceptar entregable:** recibe por parámetro el identificador del entregable a aceptar.
 - Con el identificador, se obtiene el nombre del entregable y actualiza su estado mediante la función “`acceptarEntregable(id_entregable, titulo, id_usuario, id_proyecto)`”.
 - A continuación genera el aviso automático dirigido a todos los usuarios con el rol Project Manager del proyecto actual mediante la función “`crearAviso(id_proyecto, id_emisor, id_pm, mensaje)`”.
 - Finalmente, devuelve el resultado de las funciones al fichero [entregables.php](#).
- **Rechazar entregable:** recibe por parámetro el identificador del entregable a rechazar y la razón del rechazo si ha sido indicada.
 - Con el identificador, se obtiene el nombre del entregable y actualiza su estado mediante la función “`rechazarEntregable(id_entregable, titulo, id_usuario, id_proyecto)`”.
 - A continuación genera el aviso automático, incluyendo la razón del rechazo, dirigido a todos los usuarios con el rol Project Manager del proyecto actual mediante la función “`crearAviso(id_proyecto, id_emisor, id_pm, mensaje)`”.
 - Finalmente, devuelve el resultado de las funciones al fichero [entregables.php](#).

4.19. Módulo Gestión Eliminar

A continuación se especifican los ficheros que componen el módulo “Gestión Eliminar”. Por cada fichero se explicará la funcionalidad desarrollada así como la utilización de dicha funcionalidad por parte del usuario final.



4.19.1 `gestion_eliminar.php`

Este fichero se encarga de eliminar el elemento indicado. Para ello, el fichero recibe tres parámetros: el tipo de elemento a eliminar (requisito, caso de uso, entregable, iteración o proyecto), el identificador del elemento a eliminar y el nombre del fichero a donde enviar el resultado de la operación (requisitos.php, iteraciones.php, etc.). Las funciones que realiza son las siguientes:

- **Eliminar un proyecto:** Mediante la función “eliminarProyecto(id)”.
- **Eliminar un requisito:** Mediante la función “eliminarRequisito(id)”.
- **Eliminar un caso de uso:** Mediante la función “eliminarCasoUso(id)”.
- **Eliminar una iteración:** Mediante la función “eliminarIteracion(id)”.
- **Eliminar un entregable:** Mediante la función “eliminarEntregable(id)”.

4.20. Funciones auxiliares

A lo largo de la sección actual, han aparecido numerosas funciones que realizan tareas muy concretas. Todas estas funciones están implementadas en tres ficheros: funciones.php, funciones_install.php y funciones_doc.php.

El primero de ellos contiene todas las funciones necesarias para asegurar el correcto funcionamiento de la aplicación excepto la generación de entregables. Esta tarea está implementada en el fichero funciones_doc.php. El fichero funciones_install.php contiene las funciones necesarias para llevar a cabo la instalación inicial del sistema.

Todas las funciones se encuentran especificadas en el Anexo “[Funciones auxiliares](#)”.

5. Conclusiones

La realización del proyecto ha permitido extrapolar una serie de conclusiones y posibles trabajos futuros que se exponen a continuación.

5.1. Conclusiones generales

Como se expone en la sección “[1.2. Objetivo](#)”, el principal objetivo de este proyecto era el análisis, diseño e implementación de una plataforma distribuida que permitiese a los equipos de desarrollo de proyectos software unificar toda la información que surja durante todas las fases del desarrollo para mantener un control absoluto del proyecto, de principio a fin. A lo largo del documento se ha podido seguir el desarrollo y la consecución de este objetivo con la elaboración completa de dicha aplicación.

De igual forma, se han alcanzado, también se han logrado todos los objetivos secundarios propuestos e incluso otros no propuestos inicialmente:

- Desarrollar una aplicación web que permita:
 - La gestión de proyectos de forma completa, orientando los desarrollos a la metodología [RUP](#).
 - Controlar todas las actividades o tareas que se llevan a cabo sobre un proyecto.
 - Clasificar a los participantes de un proyecto dependiendo del rol que ejerzan sobre el mismo.
 - Gestionar el Plan de Iteraciones de un proyecto.
 - Gestionar los casos de uso y requisitos de un proyecto, así como la trazabilidad entre ambos.
 - Gestionar los productos o entregables de un proyecto.
 - Publicar los elementos de un proyecto a los usuarios con el rol de propietario del proyecto en desarrollo.
 - Mantener un contacto directo entre todos los participantes del desarrollo.
 - Aceptar o rechazar elementos publicados e incluso dar la oportunidad de indicar las razones de los rechazos.
 - Personalizar la visualización del estado de cada proyecto a cada usuario.
- Desarrollar una aplicación remota que pueda trabajar sobre cualquier sistema operativo (siempre que cumpla unos requisitos mínimos) que asiente una base para futuros trabajos relacionados.

- Se ha creado una aplicación totalmente online, y en su desarrollo se ha tenido siempre en cuenta la necesidad de ser una aplicación extensible y reutilizable para trabajos futuros.
- Se ha diseñado y creado la aplicación de una forma modular que facilitará la creación de nuevas versiones y ampliaciones.
- Conocer y utilizar una gran parte de la inmensa funcionalidad que ofrece [HTML](#), [CSS](#), [PHP](#) y [MySQL](#).
- Poner en práctica los conocimientos adquiridos en la especialidad “Ingeniería de computadores” en cuanto a plataformas distribuidas y colaborativas.

Como se puede observar, se han cumplido exitosamente todos los objetivos propuestos inicialmente, documentando todo este proceso, y dejando abierta la posibilidad para la continuación de este proyecto con trabajos futuros que se verán en detalle en la siguiente sección.

5.2. Trabajos futuros

Gracias a la versatilidad que ofrece esta aplicación, permite un gran margen de maniobra, ofreciendo la posibilidad de numerosas ampliaciones, cambios y mejoras. El fin principal de la aplicación es mantener el control del desarrollo de proyectos de una forma totalmente remota y de la manera más cómoda para el usuario. Una de las primeras ideas sería ofrecer al usuario un control sobre un mayor número de elementos que aparecen en el desarrollo de un proyecto:

- Modelos de análisis.
- Prototipos de diseño.
- Partes de código.
- Informes intermedios.
- Eventos.
- Presupuestos.

Así mismo, también se podría incrementar el número de acciones para los elementos ya existentes:

- Generar un diagrama de caso de uso online integrando herramientas como [Graphviz](#) o [Gliffy.com](#) [11].
- Realizar acciones para un conjunto seleccionable de elementos.
- Edición y visualización de los entregables de forma online.

- Ampliación del número de tipos de entregable cuya generación sea automática.

Sin embargo, existen trabajos mucho más interesantes y a la vez complicados que mejorarían considerablemente la aplicación. Como ya se ha mencionado, el objetivo principal es alcanzar un control prácticamente absoluto en todas las tareas y elementos que rodean al desarrollo de un proyecto, por lo que una posibilidad consistiría en relacionar el trabajo de cada usuario con el porcentaje del desarrollo del proyecto. De esta forma la aplicación sería capaz de distribuir el trabajo restante entre los participantes de un proyecto y controlar el tiempo estimado en cada gestión de elementos como requisitos, casos de uso o incluso el código final.

Esta es una tarea muy complicada, pero sin duda supondría una gran mejora para la aplicación.

5.3. Presupuesto

El coste del análisis, diseño y desarrollo del proyecto asciende a 19.361,00€. El presupuesto se encuentra desglosado en el [Anexo I: Presupuesto](#).

6. Glosario

.NET: .NET es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware, permitiendo un rápido desarrollo de aplicaciones.

.zip: es un formato de almacenamiento sin pérdidas, muy utilizado para la compresión de datos como documentos, imágenes o programas.

Access: Microsoft Access es un sistema de gestión de bases de datos relacionales para los sistemas operativos Microsoft Windows, desarrollado por Microsoft y orientado a ser usado en un entorno personal o en pequeñas organizaciones. Es un componente de la suite ofimática Microsoft Office. Permite crear ficheros de bases de datos relacionales que pueden ser fácilmente gestionadas por una interfaz gráfica simple.

AJAX: acrónimo de Asynchronous JavaScript And XML, es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente mientras se mantiene la comunicación asíncrona con el servidor en segundo plano.

Android: es un sistema operativo móvil basado en Linux enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas y otros dispositivos.

Apache: es un servidor web HTTP de código abierto, para plataformas Linux, Microsoft Windows y Macintosh entre otras.

Array: una matriz o vector (en inglés array) es una zona de almacenamiento continuo, que contiene una serie de elementos del mismo tipo. Desde el punto de vista lógico, una es un conjunto de elementos ordenados en fila (o filas y columnas si tuviera dos dimensiones).

ASP .NET: es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML.

Base64: es un sistema de numeración posicional que usa 64 dígitos como base. Es la mayor potencia de dos que puede ser representada usando únicamente los caracteres imprimibles de ASCII. Esto ha propiciado su uso para codificación de correos electrónicos y otras muchas aplicaciones. Todas las variantes que se conocen con el nombre de Base64 usan el rango de caracteres A-Z, a-z y 0-9 en este orden para los primeros 62 dígitos, pero los símbolos escogidos para los últimos dos dígitos varían considerablemente de unas a otras.

Brute Force: En criptografía, se denomina ataque de fuerza bruta a la forma de recuperar una clave probando todas las combinaciones posibles hasta encontrar aquella que permite el acceso.

BSD: La licencia BSD es la licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Esta licencia tiene menos restricciones en comparación con otras como GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

C++: es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue extender al exitoso lenguaje de programación C con mecanismos que permitieran la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, C++ es un lenguaje híbrido.

C#: es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET. Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.

CheckBox: una casilla de verificación (o check box o checkbox o tickbox o tick box) es un elemento de interacción de la interfaz gráfica de una aplicación con el usuario, que permite a éste hacer selecciones múltiples de un conjunto de opciones.

Clave ajena: es una limitación referencial entre dos tablas. La clave ajena identifica una columna o grupo de columnas en una tabla (tabla hija o referendo) que se refiere a una columna o grupo de columnas en otra tabla (tabla maestra o referenciada). Las columnas en la tabla hija deben ser la clave primaria u otra clave candidata en la tabla referenciada.

Cliente-servidor: La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, que le da respuesta.

Cloud computing: concepto conocido como servicios en la nube o informática en la nube, es un paradigma que permite ofrecer servicios de computación a través de internet. De este modo los usuarios que acceden a los servicios disponibles “en la nube de internet” no tienen la necesidad de establecer configuraciones o instalaciones en sus máquinas locales y no tienen por qué conocer la gestión de los recursos que se utilizan.

Código abierto: Código abierto es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones éticas y morales las cuales destacan en el llamado software libre.

Common Language Runtime: El Common Language Runtime o CLR ("entorno en tiempo de ejecución de lenguaje común") es un entorno de ejecución para los códigos de los programas que corren sobre la plataforma Microsoft .NET. El CLR es el encargado de compilar una forma de código intermedio llamada Common Intermediate Language (CIL, anteriormente conocido como MSIL, por Microsoft Intermediate Language), al código de máquina nativo mediante un compilador en tiempo de ejecución.

CSS: CSS (Cascading Style Sheets) es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

CSS3: Es la última versión publicada de CSS. CSS3 está dividida en varios documentos separados, llamados "módulos". Cada módulo añade nuevas funcionalidades a las definidas en versiones anteriores, de manera que se preservan las anteriores versiones para mantener la compatibilidad.

DBSchema: es una herramienta que de manera gráfica e intuitiva ayuda durante todo el proceso de diseño, creación y posterior mantenimiento de bases de datos relacionales.

Facebook: es un sitio web de redes sociales creado por Mark Zuckerberg y fundado por Eduardo Saverin, Chris Hughes, Dustin Moskovitz y Mark Zuckerberg. Originalmente era un sitio para estudiantes de la Universidad de Harvard, pero actualmente está abierto a cualquier persona que tenga una cuenta de correo electrónico. Los usuarios pueden participar en una o más redes sociales, en relación con su situación académica, su lugar de trabajo o región geográfica.

Framework: En el desarrollo de software, un framework o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

FTP: FTP (siglas en inglés de File Transfer Protocol, “Protocolo de Transferencia de Archivos”) es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP (Transmission Control Protocol), basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

g++: es el alias tradicional de GNU C++, un conjunto gratuito de compiladores de C++.

Gb: Un gigabit es una unidad de medida de almacenamiento informático equivalente a 2^{30} bits.

Gestión de elementos: La palabra “gestión”, mencionada numerosas ocasiones a lo largo del presente documento, hace referencia a las acciones de lectura, creación, edición y borrado.

GET: Variable superglobal proporcionada por PHP, encargada de proporcionar variables de usuario al script correspondiente por medio de HTTP GET.

Gliffy.com: es una plataforma web que permite a los usuarios realizar diagramas de varios tipos de forma online. La aplicación permite a los usuarios exportar los diagramas una vez finalizados.

GPL: La Licencia Pública General de GNU o más conocida por sus siglas del inglés GNU GPL, es una licencia creada por la Free Software Foundation en 1989 y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y así protegerlo de intentos de apropiación que restrinjan estas libertades a los usuarios.

Graphviz: Graphviz (Graph Visualization Software), es un paquete de herramientas gratuito que permite generar una amplia variedad de diagramas a partir de código fuente. El paquete también proporciona herramientas para integrar el generador de diagramas en cualquier otra aplicación.

Hash (tabla): Una tabla hash es una estructura de datos que asocia llaves o claves con valores. La operación principal que soporta de manera eficiente es la búsqueda: permite el acceso a los elementos almacenados a partir de una clave generada.

HipHop for PHP: complemento creado por Facebook que permite convertir código PHP en código C++, para posteriormente ser compilado con g++ para obtener así un código objeto mucho más eficiente en consumo de recursos de lo que era el código PHP original.

HTML: HTML, siglas de “HyperText Markup Language”, es el lenguaje predominante para la elaboración de páginas web. Es utilizado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos (como imágenes). HTML se escribe en forma de “etiquetas”, rodeadas por corchetes angulares (<,>).

HTML4: HTML 4 desarrolla el lenguaje HTML con mecanismos para hojas de estilo, ejecución de scripts, marcos, objetos incluidos, soporte mejorado para texto de derecha a izquierda y direcciones mezcladas, tablas más ricas y mejoras en formularios, ofreciendo mejoras de accesibilidad para personas con discapacidades.

IIS: Internet Information Services o IIS es un servidor web y un conjunto de servicios para el sistema operativo Microsoft Windows. Este servicio convierte a un PC en un servidor web.

iOS: es un sistema operativo móvil de Apple. Originalmente desarrollado para el dispositivo iPhone, siendo después usado en dispositivos como iPod Touch, iPad y Apple TV.

Java: es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de la sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

JScript: es la implementación de Microsoft de ECMAScript. Está disponible mediante Internet Explorer y Windows Scripting Host. La versión más reciente es JScript .NET, que está basado en la versión 4 del estándar ECMAScript, y puede ser compilado para la plataforma Microsoft .NET.

JSP: JavaServer Pages (JSP) es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. Esta tecnología es un desarrollo de la compañía Sun Microsystems. Las JSP's permiten la utilización de código Java mediante scripts.

JVM: Una Máquina Virtual Java es una máquina virtual de proceso nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial (el bytecode Java), el cual es generado por el compilador del lenguaje Java.

Linux: Linux es un núcleo libre de sistema operativo basado en Unix. Es uno de los principales ejemplos de software libre. Linux está licenciado bajo la GPL v2 y está desarrollado por colaboradores de todo el mundo.

Lisp: LISt Processing. Lenguaje de programación específico para inteligencia artificial. La versión original, Lisp 1, fue inventada por John McCarthy a fines de los años 50.

LOPD: La Ley Orgánica 15/1999 de 13 de diciembre de Protección de Datos de Carácter Personal, (LOPD), es una Ley Orgánica española que tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor, intimidad y

privacidad personal y familiar. Su objetivo principal es regular el tratamiento de los datos y ficheros, de carácter personal, independientemente del soporte en el cual sean tratados, los derechos de los ciudadanos sobre ellos y las obligaciones de aquellos que los crean o tratan.

Mac/MacOS: MacOS (del inglés Macintosh Operating System) es el nombre del sistema operativo creado por Apple para su línea de computadoras Macintosh. Es conocido por haber sido el primer sistema dirigido al gran público en contar con una interfaz gráfica compuesta por la interacción del ratón con ventanas, iconos y menús.

Mbits: El megabit (Mbit o Mb) es una unidad de medida de información muy utilizada en las transmisiones de datos de forma telemática. Un megabit es equivalente a 2^{20} bits.

MD5: En criptografía, MD5 (abreviatura de Message-Digest Algorithm 5) es un algoritmo de reducción criptográfico de 128 bits. Su función se limita a una sola dirección, impidiendo el descifrado de un texto previamente cifrado.

Métrica v3: es una metodología de planificación, desarrollo y mantenimiento de sistemas de información, promovida por el Ministerio de Administraciones Públicas de España para sistemizar las actividades del ciclo de vida de los proyectos software dentro de las Administraciones Públicas.

Microsoft Office: es una suite de oficina que abarca e interrelaciona aplicaciones de escritorio, servidores y servicios para los sistemas operativos Microsoft Windows y MacOS.

Microsoft Windows: Microsoft Windows es el nombre de una familia de sistemas operativos desarrollados por Microsoft desde 1981, año en que el proyecto se denominaba “Interface Manager”. Las versiones más recientes de Windows son Windows 8 para equipos de escritorio, Windows Server 2011 para servidores y Windows Phone para dispositivos móviles.

Microsoft Windows 200: Windows 2000 es un sistema operativo de Microsoft que se puso en circulación el 17 de febrero de 2000 con un cambio de nomenclatura para su anterior sistema NT. Así, Windows NT 5.0 pasó a llamarse Windows 2000. Fue sucedido por Windows XP para equipos de escritorio en octubre de 2001 y Windows Server 2003 para servidores en abril de 2003.

MVC: Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos.

MySQL: es un sistema de gestión de bases de datos relacional, multihilo y multiusuario desarrollado por MySQL AB como software libre en un esquema de licenciamiento dual.

ODBC: Open DataBase Connectivity (ODBC) es un estándar de acceso a bases de datos desarrollado por SQL Access Group en 1992. El objetivo de ODBC es hacer posible el acceso a cualquier dato desde cualquier aplicación, sin importar qué sistema de gestión de bases de datos almacene los datos.

Open Source: En español, código abierto, es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones éticas y morales las cuales destacan en el llamado software libre.

ORM: ORM (Object-Relational Mapping), es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia.

Perl: lenguaje de programación diseñado por Larry Wall en 1987. Perl toma características del lenguaje C, del lenguaje Lisp y, en un grado inferior, de muchos otros lenguajes de programación.

PHP: PHP (PHP Hypertext Pre-processor) es un lenguaje de programación interpretado (Lenguaje de alto rendimiento), diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor aunque puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando bibliotecas externas.

php.ini: es un archivo de texto con un gran número de variables que permiten personalizar el funcionamiento de PHP en el servidor web donde se encuentre instalado.

POST: Variable superglobal proporcionada por PHP, encargada de proporcionar variables de usuario al script correspondiente por medio de HTTP GET.

Python: es un lenguaje de programación de alto nivel cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.

RadioButton: botón de opción (o radio button) es un elemento de interacción de la interfaz gráfica de una aplicación con el usuario, que permite a éste hacer una única selección de un conjunto de opciones.

RAM: La memoria de acceso aleatorio (en inglés random-access memory, cuyo acrónimo es RAM) es un conjunto de chips que junto con el microprocesador es parte fundamental del ordenador. Se trata de una memoria volátil ya que los datos almacenados en ella se pierden al apagar el ordenador.

RUP: Rational Unified Process o Proceso Unificado de Rational es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Script de base de datos: es un archivo de texto plano que almacena un conjunto de directivas que se ejecutarán en el gestor de bases de datos correspondiente. De esta forma, desde un único fichero se puede crear una base de datos completa.

Singleton: Es un patrón de diseño cuya función es restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto.

SmallTalk: es un lenguaje de programación que permite realizar tareas de computación mediante la interacción con un entorno de objetos virtuales. Metafóricamente, se puede considerar que un Smalltalk es un mundo virtual donde viven objetos que se comunican mediante el envío de mensajes.

SQL: SQL (Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre los datos. Una de sus características es el manejo del álgebra y del cálculo relacional permitiendo efectuar consultas con el fin de recuperar información de interés de una base de datos, así como también hacer cambios sobre la misma.

SQL Injection: es un método de infiltración de código intruso que aprovecha una vulnerabilidad de código presente en una aplicación a la hora de validar las entradas para realizar consultas a una base de datos.

Stakeholder: En la gestión de proyectos, los involucrados o interesados ("stakeholders" en inglés) son todas aquellas personas u organizaciones que afectan o son afectadas por el proyecto, ya sea de forma positiva o negativa.

Sun Microsystems: Sun Microsystems fue una empresa informática que se dedicaba a vender ordenadores, componentes informáticos, software y servicios informáticos. Fue adquirida en el año 2009 por Oracle Corporation, anteriormente parte de Silicon Valley, fabricante de semiconductores y software.

Symbian: es un sistema operativo que fue producto de la alianza de varias empresas de telefonía móvil.

TB: Un terabyte es una unidad de almacenamiento de información cuyo símbolo es el TB, y equivale a 2^{40} bytes.

Texto plano: es un archivo informático compuesto únicamente por texto sin formato, únicamente formado por caracteres usuales, haciéndolo legible por humanos.

Trigger: Un trigger (o disparador) en una base de datos, es un procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación. Dependiendo de la base de datos, los triggers pueden ser de inserción (INSERT), actualización (UPDATE) o borrado (DELETE).

UML: UML ("Unified Modeling Language") es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales como procesos de negocio, funciones del sistema, expresiones de lenguajes de programación, esquemas de bases de datos o componentes reutilizables.

Visual Basic: es un lenguaje de programación dirigido por eventos, desarrollado por Alan Cooper para Microsoft. Este lenguaje de programación simplifica la programación utilizando un ambiente de desarrollo completamente gráfico que facilita la creación de interfaces gráficas y, en cierta medida, la propia programación.

Windows: Microsoft Windows es el nombre de una familia de sistemas operativos desarrollados por Microsoft desde 1981, año en que el proyecto se denominaba "Interface Manager".

XML: XML, ("eXtensible Markup Language"), es un lenguaje desarrollado por el World Wide Web Consortium (W3C). Permite definir la gramática de lenguajes específicos. XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.

7. Referencias

- [1] George, S. (2004). *Advanced PHP Programming*. Sams Publishing.
- [2] *Comparison of project-management software*. Obtenido de Wikipedia: http://en.wikipedia.org/wiki/Comparison_of_project-management_software
- [3] *Información de PHP*. Obtenido de Wikipedia: <http://en.wikipedia.org/wiki/Php>
- [4] *Manual de PHP*. Obtenido de PHP.net: <http://php.net/manual/es/index.php>
- [5] *Información de PHPProjekt*. Obtenido de Wikipedia: <http://en.wikipedia.org/wiki/PHPProjekt>
- [6] *Información de Microsoft Project*. Obtenido de Wikipedia: http://en.wikipedia.org/wiki/Microsoft_project
- [7] *Información de IBM Rational Doors*. Obtenido de IBM: <http://www-142.ibm.com/software/products/es/es/ratidoor/>
- [8] *Información de IBM Rational RequisitePro*. Obtenido de IBM: <http://www-01.ibm.com/software/awdtools/reqpro/>
- [9] *Comparison of web application frameworks*. Obtenido de Wikipedia: http://en.wikipedia.org/wiki/Comparison_of_web_application_frameworks
- [10] *Comparison of relational database management systems*. Obtenido de Wikipedia: http://en.wikipedia.org/wiki/Comparison_of_relational_database_management_systems
- [11] *Herramienta de generación de diagramas*. <http://www.gliffy.com/>
- [12] *NASA Systems Engineering Handbook SP-610S*. Enero 1990. Appendix B.7—Techniques of Functional Analysis: Functional Flow Block Diagrams. <http://snebulos.mit.edu/projects/reference/NASA-Generic/NASA-STD-8739-8.pdf>

Anexo I: Presupuesto.



UNIVERSIDAD CARLOS III DE MADRID
Escuela Politécnica Superior

PRESUPUESTO DE PROYECTO

1.- Autor:

Adrián Hontoria Sánchez

2.- Departamento:

Área de Arquitectura y Tecnología de Computadores

3.- Descripción del Proyecto:

- Título: Diseño de una plataforma distribuida para el control de proyectos software.
- Duración (meses): 3
Tasa de costes Indirectos: 20%

4.- Presupuesto total del Proyecto (valores en Euros):

19.361,00 Euros

5.- Desglose presupuestario (costes directos)

PERSONAL

Apellidos y nombre	N.I.F. (no rellenar - solo a título informativo)	Categoría	Dedicación (hombres mes) ^{a)}	Coste hombre mes	Coste (Euro)
Hontoria Sánchez, Adrián		Analista	1	6.562,50	6.562,50
Hontoria Sánchez, Adrián		Diseñador	1	5.250,00	5.250,00
Hontoria Sánchez, Adrián		Programador	1	3.937,50	3.937,50
					0,00
					0,00
Hombres mes 3				Total	15.750,00

^{a)} 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)
Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

EQUIPOS

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable ^{d)}
Ordenador Intel Dual Core E5400	480,00	100	3	60	24,00
Servidor de pruebas	600,00	100	3	60	30,00
					0,00
					0,00
					0,00
					0,00
Total					54,00

^{d)} Fórmula de cálculo de la Amortización:

$$\frac{A}{B} \times C \times D$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado
B = periodo de depreciación (60 meses)
C = coste del equipo (sin IVA)
D = % del uso que se dedica al proyecto (habitualmente 100%)

SUBCONTRATACIÓN DE TAREAS

Descripción	Empresa	Coste imputable
Total		0,00

OTROS COSTES DIRECTOS DEL PROYECTO^{e)}

Descripción	Empresa	Costes imputable
Internet	Jazztel	90,00
Luz	Iberdrola	100,00
Microsoft Office 2010 estudiantes	Microsoft	140,00
Total		330,00

^{e)} Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas, otros,...

6.- Resumen de costes

Presupuesto Costes Totales	Presupuesto Costes Totales
Personal	15.750
Amortización	54
Subcontratación de tareas	0
Costes de funcionamiento	330
Costes Indirectos	3.227
Total	19.361

Anexo II: Manual de instalación.

En la presente sección se proporcionan las instrucciones necesarias para instalar la aplicación en un servidor.

1. Requisitos de arquitectura

Para poder instalar y utilizar el programa, es necesario disponer de los siguientes elementos:

- Servidor PHP 5.0 como mínimo.
- Servidor MySQL 5.0 como mínimo.
- Base de datos creada (preferiblemente vacía) y con acceso total a la misma mediante usuario y contraseña.
- Navegador a internet. (Para un resultado óptimo, utilice el navegador Google Chrome).

2. Requisitos de configuración

Debe tener activadas las siguientes opciones en el fichero “[php.ini](#)” de su servidor:

- register_globals
- register_argc_argv
- register_long_arrays
- short_open_tag

La configuración de su servidor debe permitir el uso de sesiones.

3. Instalación de cps

Siga los siguientes pasos para realizar la instalación del programa:

- Copie todo el contenido de la carpeta "CPS" en una carpeta de su servidor.
- Anote la ruta de dicha carpeta (p.e. <http://www.miservidor.com/aplicaciones/CPS/>).

- Acceda desde un navegador al fichero `install.php` (p.e. <http://www.miservidor.com/aplicaciones/CPS/install.php>).
- Siga los pasos del instalador que le aparecerá en pantalla.

Es muy importante que tome las siguientes medidas de seguridad:

- Tras finalizar la instalación, borre inmediatamente el fichero `install.php` de su servidor, ya que cualquier persona podría acceder al mismo y reiniciar la instalación. Esto conllevaría a un borrado completo de la base de datos.
- Una vez instalado el programa y borrado el fichero `install.php`, podrá acceder a la aplicación desde el fichero `index.php`.
- Ejemplo de acceso: <http://www.miservidor.com/aplicaciones/CPS/index.php> o simplemente <http://www.miservidor.com/aplicaciones/CPS>.

4. Problemas encontrados

Si ha instalado el programa correctamente y desde la página inicial no consigue acceder con su nombre de usuario y contraseña, establezca los valores de configuración de su servidor manualmente en el fichero `"/funciones/funciones.php"`.

Anexo III: Manual de usuario.

En esta sección se proporcionan las instrucciones necesarias para usar todas las funcionalidades de la aplicación.

Gestión de proyectos

Introducción a CPS

CPS (Control Distribuido de Proyectos Software) es una plataforma distribuida que permite disponer de la información directamente relacionada con el desarrollo de un proyecto desde cualquier dispositivo con acceso a internet. El programa CPS permite llevar a cabo una amplia serie de acciones cuyo objetivo principal es controlar el desarrollo de un proyecto de una forma simplificada y unificada.

CPS forma parte de un proyecto global, el cual se divide en dos ramas: gestión de usuarios y gestión de proyectos. Ésta última es abarcada completamente por CPS.

Los usuarios que pertenezcan a la primera rama, tendrán acceso a CPS dependiendo de su perfil. En dicho perfil encontrarán su rol general, los proyectos en los que participa y el rol específico para cada proyecto. CPS hace uso de dicha información para clasificar así diferentes opciones y áreas dependiendo del usuario que accede al sistema.

Los usuarios, dependiendo del rol que tengan pre-establecido, podrán crear proyectos, editarlos, gestionar requisitos, casos de uso, gestionar la trazabilidad, crear entregables, etc., todo ello, dentro del mismo entorno. Además, gracias a la lectura del perfil de usuario que hace CPS, un equipo de proyecto podrá desarrollar el proyecto teniendo un contacto directo con el cliente, ya que CPS dispone de un módulo de publicaciones, de tal forma que el cliente participe de una forma directa en el desarrollo aceptando o rechazando diferentes elementos publicados. El flujo de los datos, desde su creación hasta su aceptación, cancelación o borrado, los encontrará en la sección "[Estados de casos de uso y requisitos](#)" y en la sección "[Estados de entregables](#)".

Todo desarrollo de proyecto debe seguir una metodología para asegurar un mínimo de calidad y consistencia. CPS orienta los desarrollos mediante la metodología RUP (Rational Unified Process). Dicha metodología proporciona un enfoque disciplinado para la asignación de tareas y responsabilidades dentro de una organización de desarrollo, con el fin de garantizar la producción de software de alta calidad que satisfaga las necesidades de sus usuarios finales. CPS hace uso de dicha asignación para así tener un control absoluto de los elementos que van dando forma al proyecto a desarrollar.

Pero no significa que el desarrollador quede obligado a utilizar la metodología RUP. CPS dispone de un amplio abanico de posibilidades totalmente abiertas para no forzar el desarrollo.

Acceso a CPS

Para acceder al programa, debe dirigirse al fichero index.php de la ruta donde se haya instalado CPS. En dicha pantalla observará un formulario en el que deberá proporcionar su nombre de usuario y contraseña. Si no dispone de los de acceso, deberá consultarlo con el Administrador del sistema.



Ilustración 38. Acceso al sistema.

Tras introducir los datos, deberá pulsar el botón "Acceder" para visualizar el menú principal de CPS. Si no ha podido acceder, verifique que los datos introducidos son correctos.

En el menú principal observará tres botones: "Gestión de Proyectos", "Personalización" y "Ayuda".



Ilustración 39. Menú principal.

En la barra superior, observará que no tiene seleccionado ningún proyecto y a continuación podrá ver su rol general. Más a la derecha, verá su nombre y apellidos. Si pulsa sobre ellos, podrá acceder a la pantalla "Avisos" o salir del sistema.



Crear un proyecto

Tras seleccionar la opción "Gestión de Proyectos" accederá a la lista de proyectos creados. Dicha lista estará vacía si no existiese ninguno.



Para crear un nuevo proyecto debe pulsar en el botón superior "Nuevo proyecto". Si no visualiza dicha opción, significa que no dispone de los permisos necesarios. Consulte con su Administrador.

En la pantalla de creación de proyecto, deberá indicar el título y una pequeña descripción del proyecto a crear. Ambos campos son obligatorios. Tras proporcionar la información, haga clic en Aceptar para crear el nuevo proyecto.



Ilustración 40. Crear un proyecto.

El proyecto ya ha sido creado, pero está totalmente "vacío". Lo primero que debe hacer es crear un Plan de Iteraciones inicial.

Plan de iteraciones inicial

El Plan de Iteraciones inicial es el segundo paso de la creación de un proyecto. Como ya se ha explicado en "[Introducción a CPS](#)", el sistema orienta a los usuarios a desarrollar el proyecto siguiendo la metodología RUP, aunque no es obligatorio. Si no desea seguir dicha metodología, puede no crear ningún Plan y acceder a la pantalla principal del proyecto seleccionando el enlace "Volver".

Una de las características principales de la metodología RUP es que proporciona un desarrollo iterativo e incremental. Esto quiere decir que el desarrollo del proyecto se realiza en iteraciones. Para cada iteración o ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un mini proyecto cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo. En concreto RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié dependiendo de las exigencias del proyecto.

Para más información acerca de la creación de Iteraciones, Fases y Productos, acceda a la sección "[Plan de Iteraciones](#)".

Una vez haya acabado de realizar el plan inicial, puede hacer clic en "Volver" o bien seleccionar el proyecto desde el desplegable superior para acceder a la pantalla principal del proyecto.

Pantalla principal del proyecto

Desde esta pantalla podrá controlar todo lo relacionado con el desarrollo del proyecto. Lo primero que verá la primera vez que acceda, será la frase "No tiene activada ninguna gráfica. Puede hacerlo desde la sección Personalización.". Como bien se indica, en esa parte de la pantalla aparecerán diferentes gráficas que mostrarán el estado del proyecto. Para ello debe activarlas desde la sección "[Personalización](#)". Puede que alguna gráfica no aparezca pese a estar activada. Esto se debe a que no hay suficientes datos creados como para mostrar el contenido de la gráfica correspondiente.



Ilustración 41. Pantalla principal de un proyecto.

En la parte inferior a la sección de las gráficas, dispone de cuatro botones que le permitirán acceder a las diferentes sub-secciones del programa. Estos son: "Plan de Iteraciones", "Requisitos", "Trazabilidad", "Entregables".

Más abajo observará el estado incremental actual del proyecto. Podrá ver la fase actual y la iteración actual y podrá pasar de fase o de iteración si las restricciones del Plan de Iteraciones se lo permiten.

En la parte inferior de la pantalla, dependiendo de los permisos con los que cuente, tendrá disponible tres opciones más: "Usuarios", "Propiedades" e "Historial".

Para terminar, desde el desplegable superior central, podrá acceder a las pantallas principales de los diferentes proyectos a los que pertenezca.

Ver Proyectos

Para ver la tabla de los proyectos a los que pertenece, puede en cualquier momento pulsar el enlace "Home", situado en la parte superior izquierda de la pantalla. Dicho enlace mostrará el menú principal, por lo que deberá seleccionar la opción "Gestión de proyectos" para ver la tabla de los mismos.

De igual forma, dispone del desplegable superior central. El desplegable le mostrará los proyectos a los que pertenece y el rol que tiene asignado en cada uno de ellos.

Usuarios de un proyecto

Esta pantalla permitirá ver qué usuarios están participando en el proyecto, añadir o eliminar usuarios al proyecto y asignarles su rol correspondiente.

Las anteriores acciones sólo podrán ser realizadas por el Administrador del sistema o por el Project Manager del proyecto. Para más información, acceda a "[Permisos](#)".



Ilustración 42. Asignación de usuarios.

En la parte izquierda de cada nombre y apellidos de los usuarios, podrá seleccionar o deseleccionar un checkbox, indicando así si el usuario pertenece o no al proyecto actual.

Para asignar un rol, basta con seleccionarlo en la lista que dispone a continuación del nombre y apellidos del usuario.

Para guardar los cambios pulse "Aceptar".

Propiedades

Desde la pantalla propiedades podrá editar el título y la descripción del proyecto.

Las anteriores acciones sólo podrán ser realizadas por el Administrador del sistema o por el Project Manager del proyecto. Para más información, acceda a "[Permisos](#)".

Para guardar los cambios que haya realizado, pulse "Aceptar".

Historial

Desde esta sección tendrá un control absoluto sobre las acciones que se realizan en el proyecto. Toda acción queda registrada en el historial, indicando quién lo ha realizado, el qué ha realizado, qué valor ha sufrido la acción y la fecha en la que se realizó.

Podrá ordenar el historial de forma ascendente o descendente haciendo clic en los iconos triangulares negros, situados debajo de cada título de cada columna.

Los enlaces "Anterior" y "Siguiente" estarán disponibles cuando exista más de una página, o lo que es lo mismo, cuando existan más de 15 filas por página.

Gráficas de un proyecto

Las gráficas de un proyecto le permitirán recibir de forma gráfica un resumen del estado actual del mismo. Recuerde que puede activar o desactivar las diferentes gráficas desde la sección "[Personalización](#)".

Las gráficas disponibles en CPS son las siguientes:

- Número de requisitos suplementarios por estado.
- Número de iteraciones por estado y fases.
- Requisitos editados por semana.
- Estabilidad media de requisitos por semana.
- Número de iteraciones por estado.
- Casos de uso por estado.
- Entregables por estado.
- Casos de uso editados por semana.
- Entregables editados por semana.
- Requisitos por versión.
- Casos de uso por versión.
- Entregables por versión.

Más detalles en la sección "[Personalización](#)".

Editar un proyecto

Esta opción estará disponible desde el listado de todos los proyectos, tras seleccionar la opción "Gestión de proyectos" en el menú principal.

La opción "Editar un proyecto" es la misma que la opción "Propiedades".

Descargar un proyecto

La opción de descarga del proyecto permitirá al usuario obtener un fichero .zip cuyo contenido son todos los entregables generados tras el desarrollo del proyecto.

Para más información, acceda a la sección "[Entregables](#)".

Plan de Iteraciones

El Plan de Iteraciones

Tal como se explicó en la sección [Gestión de proyectos](#), el sistema CPS orienta de forma opcional el desarrollo de proyectos mediante la metodología RUP (Rational Unified Process).

Una de las características principales de la metodología RUP es que proporciona un desarrollo iterativo e incremental. Esto quiere decir que el desarrollo del proyecto se realiza en iteraciones. Para cada iteración o ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un mini proyecto cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo. En concreto RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié dependiendo de las exigencias del proyecto.

Las fases en las que RUP divide el proyecto y por tanto la que CPS proporciona de forma opcional son: Inicio, Elaboración, Construcción y Transición. La distribución del trabajo en cada una de las fases depende del proyecto a realizar por lo que el programa no obliga al usuario a realizar productos específicos para cada fase. Dichos productos son los indicados por el usuario al crear cada iteración.

Resumiendo los elementos de un Plan de Iteraciones, cada una de las fases se compone de diferentes iteraciones creadas por el usuario. En cada iteración se indica un producto o productos a obtener y de forma opcional, un responsable de la realización de cada producto.

Crear nueva iteración

Para crear una nueva iteración, el usuario debe seleccionar el botón "Nueva iteración", desde la sección Plan de iteraciones.



En primer lugar, se debe seleccionar la fase de la que formará parte la iteración que se va a crear. Se debe indicar también el nombre de la iteración y la duración en semanas de la misma. Los tres campos anteriores son obligatorios.

Ilustración 43. Crear una iteración.

El siguiente paso es la asignación de productos. El usuario deberá indicar qué productos se obtendrán a lo largo de la iteración para así pasar a la siguiente iteración del Plan. La selección de productos es opcional ya que el usuario puede necesitar otro producto que no esté listado. Este caso se explica en más detalle en el punto "[Productos a obtener](#)".

Para finalizar la creación, el usuario debe hacer clic en el botón "Aceptar".

Ver iteraciones

Desde la pantalla principal de la sección "Plan de iteraciones" el usuario podrá ver una lista de todas las iteraciones creadas clasificadas por las fases a las que pertenecen.

<input type="radio"/> Fase 1: Inicio <input type="radio"/> Fase 2: Elaboración <input checked="" type="radio"/> Fase 3: Construcción <input type="radio"/> Fase 4: Transición					
Nº	Nombre	Duración (en semanas)	Estado	Productos	
1	Funcionalidades básicas	3	Pendiente	Listado	
2	Funcionalidades adicionales	3	Pendiente	Listado	
3	Gráficas y documentos	3	Pendiente	Listado	
4	Funcionalidades extra	2	Pendiente	Listado	

Ilustración 44. Ver iteraciones.

Si el usuario hace clic en el selector de una fase, se mostrarán las iteraciones de dicha fase. Por cada iteración, el usuario podrá llevar a cabo diferentes acciones:

- Ver los productos de cada iteración: Para ello el usuario debe hacer clic en el desplegable "Listado" y así poder observar los productos que se esperan obtener durante el desarrollo de la iteración seleccionada.
- Editar iteración: Puede que el Plan inicial fuese insuficiente o debido a otros problemas no se hayan conseguido los productos indicados en el tiempo planificado, de tal forma que sea necesario pasar de iteración aunque no se hayan cumplido los objetivos. CPS dispone de la opción de editar para que el usuario pueda desasignar productos o cambiar otra información de la iteración.
- Ver el historial de la iteración: el usuario podrá observar todas las acciones que se han realizado sobre la iteración seleccionada. Se indicará también el responsable de cada acción realizada.

- Eliminar iteración: Solo el administrador puede llevar a cabo esta acción. Más detalles en la sección "[Permisos](#)".

Editar iteración

En la ventana de edición de la iteración, el usuario podrá modificar el nombre de la iteración y la duración en semanas.

También tendrá la posibilidad de desasignar productos o asignar otros nuevos y cambiar el responsable de cada uno de ellos.

Para finalizar, deberá hacer clic en el botón "Aceptar".

Productos a obtener

Una de las secciones más importantes del sistema CPS es la generación de entregables, documentos que irán destinados al cliente. Dichos entregables son los productos que se han ido desarrollando en cada iteración. Es por tanto totalmente obligatorio tener terminados los productos de una iteración para poder pasar a la siguiente.

CPS controla esta situación, de tal forma que el usuario no pueda pasar de iteración o fase hasta que no se hayan generado todos los entregables correspondientes a los productos que se indicaron. Aunque existe una vía de escape para situaciones excepcionales: la desasignación de productos. El usuario tendrá la posibilidad de desasignar un producto para así poder ajustarse a las nuevas necesidades y tener la posibilidad de avanzar en el desarrollo del proyecto.

Todos los productos se gestionarán desde la sección "[Entregables](#)".

Pasar de iteración

Desde la pantalla principal del proyecto, el usuario podrá pasar de iteración haciendo clic en el botón "Pasar de iteración". Dicho botón estará visible siempre que exista una iteración siguiente y se hayan generado todos los productos de la iteración actual. Si no se cumple alguna de estas condiciones, CPS lo indica en dicha pantalla.

Pasar de fase

Desde la pantalla principal del proyecto, el usuario podrá pasar de fase haciendo clic en el botón "Pasar de fase". Dicho botón estará visible siempre que exista una iteración creada en la fase siguiente y se hayan generado todos los productos de la iteración actual. Si no se cumple alguna de estas condiciones, CPS lo indica en dicha pantalla.

Gestión de requisitos y casos de uso

Requisitos y casos de uso

Un proyecto software se crea para servir a sus usuarios. Por lo tanto, para construir un sistema de calidad se debe conocer qué es lo que los usuarios quieren y necesitan. Hay que destacar que el término "usuarios" no se refiere únicamente a los usuarios humanos, sino a otros sistemas. En este contexto, el término "usuario" representa algo o alguien que interactúa con el sistema a desarrollar.

En la metodología RUP, un caso de uso es una pieza en la funcionalidad del sistema que le da al usuario un resultado de valor. Los casos de uso capturan los requisitos funcionales. Todo el conjunto de casos de uso constituyen el modelo de casos de uso el cual describe la funcionalidad completa del sistema.

Este modelo reemplaza la tradicional especificación funcional de un sistema. Una especificación funcional tradicional se concentra en responder a la pregunta: ¿Qué debe hacer el sistema? La estrategia de casos de uso responde a una pregunta aún más específica: ¿Qué debe hacer el sistema por cada usuario? Esta estrategia fuerza a pensar en términos de valor a los usuarios y no únicamente en términos de las funciones que se relacionen con cada usuario.

Sin embargo, los casos de uso no son solamente una herramienta para especificar los requisitos del sistema, también dirigen su diseño, implementación y pruebas. Se puede afirmar por tanto que los casos de uso, a través de los requisitos entre otros elementos, dirigen el proceso de desarrollo.

Crear nuevo requisito

Desde la sección "[Requisitos](#)" el usuario podrá crear un nuevo requisito seleccionando el botón "Nuevo requisito". Accederá a la pantalla de creación de requisitos y casos de uso.



Desde dicha pantalla, el usuario deberá seleccionar el elemento a crear: "Caso de uso" o "Suplementario". Para crear un requisito, debe seleccionar "Suplementario". Tras su selección, aparecerán diferentes campos necesarios para la creación del nuevo requisito:



Requisito: Suplementario

Trazabilidad:

Título:

Descripción:

Impacto: Muy bajo

Esfuerzo: Muy bajo

Tipo:

Aceptar

Ilustración 45. Crear un requisito suplementario.



- Trazabilidad: el usuario debe seleccionar qué trazabilidad tendrá el nuevo requisito. Dispone de tres opciones:
 - Con caso de uso: significa que el nuevo requisito dispone de una dependencia directa con un caso de uso. Dicho caso de uso puede crearse antes o después de la creación del requisito. Para asignar la trazabilidad con el caso de uso, el usuario podrá hacerlo posteriormente desde la sección "Trazabilidad".
 - Global: significa que el requisito no dispone de ninguna dependencia directa con los elementos más destacados del proyecto. De este tipo suelen ser los requisitos que definen interfaces externas con otros sistemas.
 - De arquitectura: este tipo de trazabilidad indica que el requisito especificará detalles relacionados con la arquitectura del sistema en desarrollo, alejándose de la funcionalidad.
- Título: un título descriptivo del requisito.
- Descripción: adicional al título.
- Impacto: indica el efecto de dicho requisito sobre el sistema a desarrollar. Los niveles son: Muy Bajo, Bajo, Medio, Alto y Muy Alto.
- Esfuerzo: indica una estimación de forma aproximada del esfuerzo para hacer realidad dicho requisito. Los niveles son: Muy Bajo, Bajo, Medio, Alto y Muy Alto.
- Tipo: el usuario dispone de una amplia variedad de tipos de requisito. Si quiere crear un tipo nuevo, debe seleccionar "Otro" y escribir el nuevo valor en el nuevo campo de texto. Dicho valor se guardará automáticamente para futuros usos en el proyecto.
- Subtipo: dependiendo del tipo, pueden existir diferentes subtipos de requisitos. El usuario podrá seleccionar subtipos predefinidos o crear nuevos. Para ello, debe seleccionar "Añadir nuevo" y escribir el nuevo valor en el nuevo campo de texto. Dicho valor se guardará automáticamente para futuros usos en el proyecto.

Todos los campos anteriores, excepto el subtipo, son obligatorios.

Para finalizar con la creación del requisito, el usuario deberá seleccionar el botón "Aceptar".

Crear nuevo caso de uso

Desde la sección "[Requisitos](#)" el usuario podrá crear un nuevo caso de uso seleccionando el botón "Nuevo requisito". Accederá a la pantalla de creación de requisitos y casos de uso.

Desde dicha pantalla, el usuario deberá seleccionar el elemento a crear: "Caso de uso" o "Suplementario". Para crear un caso de uso, debe seleccionar "Caso de uso". Tras su selección, aparecerán diferentes campos necesarios para la creación del nuevo caso de uso:




Ilustración 46. Crear un caso de uso.

- Título: el título descriptivo del caso de uso.
- Imagen: el archivo que contiene la imagen del nuevo caso de uso. Actualmente el sistema CPS no dispone de un editor de gráficos on-line, por lo que el usuario deberá generar de forma externa el caso de uso. La imagen podrá tener alguna de estas extensiones: .jpg, .jpeg, .bmp y .png.

Todos los campos anteriores son obligatorios.

Para finalizar con la creación del caso de uso, el usuario deberá seleccionar el botón "Aceptar".

Trazabilidad entre requisito y caso de uso

Para acceder a la sección de trazabilidad, el usuario puede hacerlo desde la pantalla principal del proyecto, seleccionando la opción "Trazabilidad" o directamente desde la sección "Requisitos", pulsando el enlace "Crear" de la columna "Trazas" del listado de todos los requisitos creados.

La sección "Trazabilidad" muestra la matriz de trazabilidad entre los requisitos (creados con trazabilidad con caso de uso) y los casos de uso. Para editar la trazabilidad, el usuario debe hacer clic en el botón "Editar". Si no aparece visible dicho botón, significa que el usuario no cuenta con suficientes permisos.

	CU8: gestionar_aviso	CU7: personalizar_grafico	CU6: gestionar_entregable	CU5: gestionar_trazabilidad	CU4: gestionar_casos_de_uso	CU3: gestionar_requisitos	CU2: gestionar_iteraciones	CU1: gestionar_proyectos
R1: Gestionar proyectos								
R2: Gestionar iteraciones							✓	✓
R3: Gestionar requisitos						✓		
R4: Gestionar casos de uso					✓			
R5: Gestionar trazabilidad				✓				

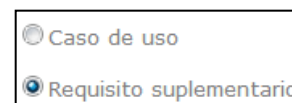
Ilustración 47. Matriz de trazabilidad.

Tras pulsar el botón "Editar", el usuario entrará en el modo de edición, desde donde podrá activar o desactivar las relaciones directamente desde la matriz de trazabilidad. Una vez establecidas las relaciones entre los requisitos y casos de uso, el usuario deberá seleccionar el botón "Guardar" y observará como sus cambios se han actualizado en la matriz de trazabilidad.

Ver requisitos y casos de uso

El usuario accederá a la pantalla principal de requisitos y casos de uso, seleccionando la opción "Requisitos" desde [la pantalla principal del proyecto](#).

La nueva pantalla mostrará por defecto un listado con todos los requisitos creados. Si el usuario desea ver el listado de los casos de uso, deberá seleccionar el botón tipo radio "Caso de uso".



<input checked="" type="radio"/> Caso de uso
<input type="radio"/> Requisito suplementario

El listado de los requisitos muestra por cada fila toda la información de cada requisito creado. Además dispone de una serie de acciones para realizar sobre el requisito seleccionado:

- ID: Es el identificador único del requisito. Dicho identificador se compone de las letras "ID" seguido de un número entero incremental. El ID es calculado automáticamente por el sistema.
- Título y descripción: los datos indicados al crear el requisito.
- Estado: el estado actual del requisito. Más información en "[Estados](#)". El usuario podrá situar el cursor del ratón sobre el icono para obtener el nombre del estado.
- Tipo, subtipo, impacto y esfuerzo: los datos indicados al crear el requisito.
- Versión: la versión es un número entero e incremental que indica el número de veces que ha sido editado un requisito. Un requisito recién creado comienza con versión 1, incrementándose de uno en uno por cada edición que sufra. La versión es calculada automáticamente por el sistema.
- Trazas: si el requisito dispone de trazabilidad con algún caso de uso, se mostrará el listado de dichas dependencias si el usuario hace clic en el desplegable. Si no dispone de ninguna traza, se muestra el texto "Crear", para que el usuario pueda definir una nueva traza.
- Editar: Esta acción permite editar la información del requisito. Más información en "Editar requisitos y casos de uso". Esta acción está sujeta a restricciones de permisos.
- Historial: Esta acción permite al usuario ver todas las acciones que se han realizado sobre el requisito seleccionado. Por cada acción el sistema muestra quién la llevó a cabo y cuándo fue realizada.
- Publicar: El usuario podrá publicar el requisito para que sea evaluado por el Project Manager o directamente por el cliente, dependiendo del rol del usuario. Esta acción está sujeta a restricciones de [permisos](#).
- Eliminar: Solo el administrador del sistema podrá realizar esta acción.

El listado de los casos de uso muestra por cada fila toda la información de cada caso de uso creado. Además dispone de una serie de acciones para realizar sobre el caso de uso seleccionado:



- ID: Es el identificador único del caso de uso. Dicho identificador se compone de las letras "ID" seguido de un número entero incremental. El ID es calculado automáticamente por el sistema.
- Título: el título indicado al crear el caso de uso.
- Estado: el estado actual del caso de uso. Más información en "Estados". El usuario podrá situar el cursor del ratón sobre el icono para obtener el nombre del estado.
- Versión: la versión es un número entero e incremental que indica el número de veces que ha sido editado un caso de uso. Un caso de uso recién creado comienza con versión 1, incrementándose de uno en uno por cada edición que sufra. La versión es calculada automáticamente por el sistema.
- Editar: Esta acción permite editar la información del caso de uso. Más información en "Editar requisitos y casos de uso". Esta acción está sujeta a restricciones de [permisos](#).
- Historial: Esta acción permite al usuario ver todas las acciones que se han realizado sobre el caso de uso seleccionado. Por cada acción el sistema muestra quién la llevó a cabo y cuándo fue realizada.
- Publicar: El usuario podrá publicar el caso de uso para que sea evaluado por el Project Manager o directamente por el cliente, dependiendo del rol del usuario. Esta acción está sujeta a restricciones de [permisos](#).
- Eliminar: Solo el administrador del sistema podrá realizar esta acción.

Editar requisitos y casos de uso





El usuario podrá editar información de cada requisito y caso de uso. Para ello, desde el listado de requisitos o casos de uso, deberá seleccionar la opción "Editar".


Desde la nueva pantalla de edición, el usuario podrá modificar los campos que se indicaron a la hora de crear el requisito o caso de uso. Si el usuario se dispone a modificar un caso de uso, dispondrá de una previsualización de la imagen que subió anteriormente, teniendo la posibilidad de modificarla subiendo una nueva.

Para guardar los cambios el usuario deberá hacer clic en el botón "Aceptar". El sistema guardará los cambios realizados y calculará la nueva versión.

Estados

Cada requisito y caso de uso dispone de un ciclo de vida representado por varios estados. Estos estados son los mismos para los requisitos y los casos de uso:

- **Procesado:** este estado indica que el requisito o caso de uso ha sido creado y está a la espera de ser publicado o editado. Desde este estado se permite realizar cualquier acción sobre el requisito o caso de uso. El estado está representado por un icono de reloj de arena .
- **Revisando:** el requisito o caso de uso pasa a este estado cuando es enviado a evaluación al Project Manager. Este estado perdurará hasta que el Project Manager publique el requisito o caso de uso al cliente. De igual forma, si un usuario edita el requisito o caso de uso, el sistema avisará con una alerta, ya que el estado volverá a "Procesado" tras su edición. El estado "Revisando" está representado por un icono de interrogación . El Project Manager recibirá un aviso de nuevo requisito o caso de uso listo para evaluar. Más información en la sección "[Avisos](#)".
- **Publicado:** el requisito o caso de uso ha sido publicado por el Project Manager al cliente. Este requisito o caso de uso pasará a estar visible en la zona del cliente para que éste pueda aceptarlo o rechazarlo. De igual forma, el cliente recibirá un aviso con la nueva publicación. Este estado perdurará hasta que el cliente acepte o rechace el requisito o caso de uso o bien un usuario edite el elemento. El estado "Publicado" está representado por un icono de un hombre con traje .
- **Rechazado:** indica que el propietario ha rechazado el requisito o caso de uso que fue publicado. Al realizar esta opción, el sistema proporciona al cliente un campo de texto para que indique las razones de su rechazo. Estas razones se incluirán en el aviso que llegará al Project Manager. Este estado perdurará hasta que el requisito o caso de uso sea editado, para volver a comenzar su ciclo de vida desde el estado "Procesado". El estado "Rechazado" está representado por un icono de una cruz roja .
- **Aceptado:** el estado indica que el propietario ha aceptado el requisito o caso de uso. Al aceptar el elemento, el sistema avisa al Project Manager con la decisión tomada. El estado "Aceptado" es el último en el ciclo de vida de un requisito o caso de uso. Indica que el elemento podrá ser añadido a los entregables finales.

correspondientes. El estado "Aceptado" está representado por un icono de "tick" verde .

Publicar requisito o caso de uso

Existen dos tipos de publicación de requisitos o casos de uso, dependiendo del rol que la realice.

La primera de ellas es realizada por los Ingenieros de Requisitos. Cuando el usuario con dicho rol seleccione la opción "Publicar" desde el listado de requisitos o casos de uso, el elemento pasará al estado "Revisando". En este momento el sistema manda un aviso al Project Manager (o a todos los Project Manager si hay varios) indicando que tiene disponible un requisito o caso de uso para evaluar. El Project Manager podrá ver el aviso desde su panel de avisos y acceder al listado de requisitos o casos de uso en cuestión. Una vez que el Project Manager revise el requisito o caso de uso, podrá tomar dos acciones: responder el aviso anterior para que el usuario realice modificaciones en el requisito o publicar el requisito o caso de uso para el/los cliente/es del sistema en desarrollo.

En el primer caso, el Project Manager responderá el aviso indicando los errores a corregir, de tal forma que el usuario responsable de mandar la evaluación del elemento, recibirá dicho aviso y procederá a modificar el requisito o caso de uso.

En el segundo caso, el Project Manager publicará el requisito o caso de uso, de tal forma que el elemento publicado cambia al estado "Publicado". En ese mismo momento, el sistema manda un aviso a todos los clientes del proyecto en desarrollo, indicando que el Project Manager ha publicado un requisito o caso de uso. Más información en la sección "[Avisos](#)".

Gestión de entregables

Entregables

Un entregable es un producto que se desea obtener de forma parcial o total al finalizar una iteración o una fase. La metodología RUP (Rational Unified Process) se refiere a estos productos mediante el término "artefactos".

RUP define los artefactos e incluso los clasifica por las fases en los que se obtienen. El sistema CPS no es tan estricto, ya que permite realizar un Plan de Iteraciones abierto, dando la posibilidad al usuario de seleccionar los productos que se esperan obtener o incluso generar productos no contemplados por RUP.

Hay que destacar que un entregable es un elemento más del proceso de desarrollo, por lo que dispondrá de un control de versiones y de diferentes estados que definirán su ciclo de vida.

Tipos de entregables

El sistema CPS proporciona de forma predeterminada una serie de productos a la hora de asignarlos al Plan de Iteraciones. Como se ha mencionado anteriormente, CPS permite una cierta flexibilidad a la hora de crear los productos, ya que el usuario podrá crear entregables totalmente diferentes a los listados por defecto.

Los productos por defecto son los siguientes:

- Plan de desarrollo: proporciona en detalle el Plan de Iteraciones con el que se está llevando a cabo el desarrollo del proyecto. Puede ser generado de forma automática.
- Caso de negocio: proporciona la lógica del negocio del proyecto. El producto incluye la entrada y salida de datos así como su procesamiento interno. Debe ser generado de forma manual.
- Lista de Riesgos: Listado de todos los riesgos que pueden afectar al sistema en desarrollo. Deben indicarse las amenazas, las consecuencias y las soluciones. El producto debe ser generado de forma manual.
- Modelo de casos de uso: proporciona todos los casos de uso aceptados del proyecto. Este producto describe la funcionalidad del sistema en desarrollo frente a los usuarios que lo utilizarán. El producto puede ser generado de forma automática.
- Visión: este producto define una descripción global del sistema desde el punto de vista de los [stakeholders](#), centrándose en las necesidades y características del futuro sistema. El producto debe ser generado de forma manual.
- Requisitos suplementarios: este producto proporciona un listado de los requisitos suplementarios que han sido aceptados. Se incluye también la matriz de trazabilidad con los casos de uso. El producto puede ser generado de forma automática.

- **Glosario:** proporciona aquellos términos utilizados en diferentes productos que sean poco conocidos o de difícil interpretación. El producto debe ser generado de forma manual.
- **Modelo de diseño:** Es un modelo físico y concreto, un plano de la implementación. Proporciona en su conjunto las clases de diseño junto con los casos de uso con los que se relacionan. El producto debe ser generado de forma manual.
- **Arquitectura Software:** involucra los aspectos estáticos y dinámicos más significativos del sistema. El producto dispone de muchos factores, tales como la plataforma de software en la que se ejecutará el sistema, la disponibilidad de componentes reutilizables, consideraciones de instalación, requisitos no funcionales, etc. El producto debe ser generado de forma manual.
- **Modelo de implementación:** consiste en una visión general acerca de lo que tiene que ser implementado. Se indican los componentes y subsistemas a implementar por cada iteración, así como el resultado que se espera obtener. El producto debe ser generado de forma manual.
- **Plan de test:** indica por cada iteración los elementos a testear, qué objetivos buscan las pruebas y una aproximación a utilizar para conseguir los objetivos. El producto debe generarse de forma manual.
- **Plan de despliegue:** incluye varios procesos de despliegue como empaquetar el software para su distribución, la distribución, la instalación, la asistencia y ayuda a los usuarios, etc. El producto debe generarse de forma manual.

Crear nuevo entregable

El usuario podrá crear un nuevo entregable seleccionando el botón "Nuevo entregable" desde la [pantalla principal de entregables](#).

Desde la nueva ventana el usuario deberá seleccionar el producto a general. Hay que destacar que el listado de productos que se muestran son los correspondientes a la iteración y fase actuales. Si el usuario necesita generar un producto diferente a los listados, puede tomar dos caminos:



- [Modificar el Plan de Iteraciones](#): el usuario deberá añadir el producto como objetivo de la iteración actual.
- Crear "Otro entregable". El último elemento en el listado de los productos que se pueden crear, el usuario tiene disponible la opción "Otro". Si selecciona esta opción, aparecerán visibles dos campos:
 - **Título:** indica el título del entregable a crear. Campo obligatorio.
 - **Descripción:** descripción detallada del nuevo entregable. Campo obligatorio.

Finalmente, tanto para productos predefinidos como para otros productos, el usuario tiene disponible un apartado para la carga de ficheros externos al sistema. Si el entregable

seleccionado no puede ser generado de forma automática, será obligatorio incluir documentos externos.

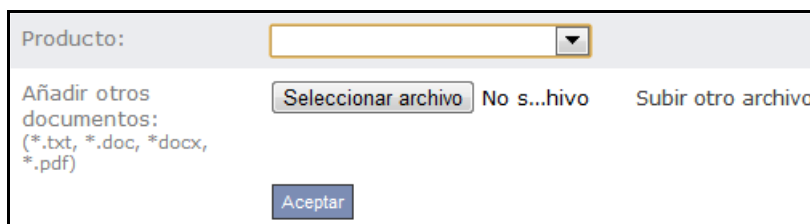


Ilustración 48. Crear un entregable.

Los documentos que pueden cargarse deben disponer de alguna de estas extensiones: .txt, .doc, .docx o .pdf.

El usuario podrá cargar tantos ficheros como desee. Para ello deberá hacer clic sobre el enlace "Subir otro archivo". Así se hará visible un nuevo campo de carga de ficheros.

Por último, el usuario deberá hacer clic en el botón "Aceptar" para generar el entregable. Dicho entregable contendrá su contenido propio, si se genera de forma automática, junto con el contenido de los documentos que se han cargado de forma externa.

Ver entregables

El usuario accederá a la pantalla principal de entregables seleccionando la opción "Entregables" desde [la pantalla principal del proyecto](#).

La nueva pantalla mostrará el listado con todos los entregables creados.

El listado de los entregables muestra por cada fila toda la información de cada entregable creado. Además dispone de una serie de acciones para realizar sobre el entregable seleccionado:

- ID: Es el identificador único del entregable. Dicho identificador se compone de las letras "ID" seguido de un número entero incremental. El ID es calculado automáticamente por el sistema.
- Título y descripción: los datos indicados al crear el entregable o los datos por defecto.
- Estado: el estado actual del entregable. Más información en "Estados". El usuario podrá situar el cursor del ratón sobre el icono para obtener el nombre del estado.
- Fase: la fase en la que se encontraba el proyecto cuando se generó el entregable.
- Iteración: la iteración en la que se encontraba el proyecto cuando se generó el entregable.
- Versión: la versión es un número entero e incremental que indica el número de veces que ha sido actualizado un entregable. Un entregable recién creado comienza con versión 1, incrementándose de uno en uno por cada edición que sufra. La versión es calculada automáticamente por el sistema.

- Actualizar: Esta acción permite actualizar el contenido del entregable. Más información en "Actualizar entregables". Esta acción está sujeta a restricciones de [permisos](#).
- Publicar: El usuario podrá publicar el entregable para que sea evaluado por el Project Manager o directamente por el cliente, dependiendo del rol del usuario. Esta acción está sujeta a restricciones de [permisos](#).
- Descargar: El usuario podrá descargarse el entregable en formato .doc.
- Eliminar: Solo el administrador del sistema podrá realizar esta acción.

Actualizar entregables

Un usuario podrá crear una nueva versión de un entregable, de tal forma que se contenga información actualizada del proyecto u otra información externa.



Existen por tanto dos tipos de actualización: la actualización de documentos que generan contenido de forma automática y la actualización de documentos que dependen por completo de la información externa que el usuario ha cargado.


En primer lugar, cuando un usuario selecciona la opción "Actualizar" en un entregable que genera contenido con la información registrada en el sistema, el entregable anterior es borrado, generando un entregable totalmente nuevo y actualizado. Si el usuario indicó en una versión anterior información externa, deberá incluirla de nuevo.



El otro modo consiste en introducir más información externa al entregable. Únicamente el usuario deberá incluir los documentos externos en el formulario de actualización y pulsar el botón "Aceptar". El sistema añadirá la nueva información al contenido anterior y generará una nueva versión del entregable.

Estados

Cada entregable dispone de un ciclo de vida representado por varios estados:

- Procesado: este estado indica que el entregable ha sido creado y está a la espera de ser publicado o editado. Desde este estado se permite realizar cualquier acción sobre el entregable. El estado está representado por un icono de reloj de arena .
- Revisando: el entregable pasa a este estado cuando es enviado a evaluación al Project Manager. Este estado perdurará hasta que el Project Manager publique el entregable al cliente. De igual forma, si un usuario actualiza el entregable, el sistema avisará con una alerta, ya que el estado volverá a "Procesado" tras su edición. El estado "Revisando" está representado por un icono de interrogación . El Project Manager recibirá un aviso de nuevo entregable listo para evaluar. Más información en la sección "[Avisos](#)".
- Publicado: el entregable ha sido publicado por el Project Manager al cliente. Este entregable pasará a estar visible en la zona del cliente para que éste pueda aceptarlo o rechazarlo. De igual forma, el cliente recibirá un aviso con la nueva publicación. Este estado perdurará hasta que el cliente acepte o rechace el

entregable o bien un usuario edite el elemento. El estado "Publicado" está representado por un icono de un hombre con traje .

- Rechazado: indica que el propietario ha rechazado el entregable que fue publicado. Al realizar esta acción, el sistema proporciona al cliente un campo de texto para que indique las razones de su rechazo. Estas razones se incluirán en el aviso que llegará al Project Manager. Este estado perdurará hasta que el entregable sea actualizado, para volver a comenzar su ciclo de vida desde el estado "Procesado". El estado "Rechazado" está representado por un icono de una cruz roja .
- Aceptado: el estado indica que el propietario ha aceptado el entregable. Al aceptar el elemento, el sistema avisa al Project Manager con la decisión tomada. El estado "Aceptado" es el último en el ciclo de vida de un entregable. El estado "Aceptado" está representado por un icono de "tick" verde .

Publicar

Existen dos tipos de publicación de entregables, dependiendo del rol que la realice.

La primera de ellas es realizada por los Responsables de Entregables. Cuando el usuario con dicho rol seleccione la opción "Publicar" desde el listado de entregables, el elemento pasará al estado "Revisando". En este momento el sistema manda un aviso al Project Manager (o a todos los Project Manager si hay varios) indicando que tiene disponible un entregable para evaluar. El Project Manager podrá ver el aviso desde su panel de avisos y acceder al listado de entregables en cuestión. Una vez que el Project Manager revise el producto, podrá tomar dos acciones: responder el aviso anterior para que el usuario realice modificaciones sobre el entregable o publicarlo para el/los cliente/es del sistema en desarrollo.

En el primer caso, el Project Manager responderá el aviso indicando los errores a corregir, de tal forma que el usuario responsable de mandar la evaluación del elemento, recibirá dicho aviso y procederá a modificar el entregable.

En el segundo caso, el Project Manager publicará el entregable, de tal forma que el elemento publicado cambia al estado "Publicado". En ese mismo momento, el sistema manda un aviso a todos los clientes del proyecto en desarrollo, indicando que el Project Manager ha publicado un entregable. Más información en la sección "Avisos".

Descargar

El usuario podrá descargarse el entregable en forma de documento con extensión .doc. Para ello basta con hacer clic en el botón "Descargar" del [listado de entregables](#).

De forma automática aparecerá la ventana de descarga de fichero, que permitirá al usuario guardar el documento en su ordenador.

Personalización

Área de personalización

El usuario podrá acceder al área de personalización desde [el menú principal](#), haciendo clic sobre la opción "Personalización" o desde [la pantalla principal de proyecto](#), concretamente desde el área de gráficas.

El área de personalización permite a cada usuario personalizar las gráficas que se mostrarán en la pantalla principal de cada proyecto al que esté asignado. Esto es de gran utilidad ya que cada usuario puede centrarse en las áreas que correspondan a su rol y tareas.

Para activar o desactivar diferentes gráficas, el usuario en primer lugar debe seleccionar el proyecto que va a personalizar de la lista. Tras su selección, se cargará la configuración actual de las gráficas respecto al proyecto seleccionado.

Proyectos: CPS

Gráficas disponibles:

- ☒ Número de requisitos suplementarios por estado
- ☒ Número de iteraciones por estado y fases
- ☒ Requisitos editados por semana
- ☐ Estabilidad media de requisitos por semana
- ☐ Número de iteraciones por estado
- ☒ Casos de uso por estado
- ☐ Entregables por estado
- ☒ Casos de uso editados por semana
- ☐ Entregables editados por semana
- ☐ Requisitos por versión
- ☒ Casos de uso por versión
- ☒ Entregables por versión

Aceptar

En este punto, el usuario puede activar o desactivar diferentes gráficas y para confirmar sus cambios debe pulsar el botón "Aceptar". La próxima vez que acceda a la pantalla principal del proyecto, podrá visualizar las gráficas seleccionadas.

Tipos de gráficas

El sistema CPS dispone de 12 gráficas referentes a elementos de los proyectos en desarrollo:

- Número de requisitos suplementarios por estado: gráfica del tipo circular en 3D que muestra los porcentajes del número de requisitos por los diferentes estados.



Ilustración 49. Ejemplo de gráfica de requisitos suplementarios por estado.

- Número de iteraciones por estado y fases: gráfica de barras en el que indica el número de iteraciones (eje Y) por cada fase (eje X). Además cada barra dispone de un color: Azul si la iteración tiene estado "Pendiente" o roja si su estado es "Cerrada".

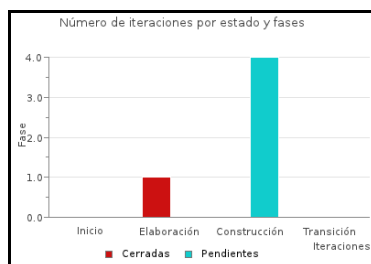


Ilustración 50. Ejemplo de gráfica iteraciones por estado y fases.

- Requisitos editados por semana: Gráfica de barras que indica el número de ediciones sobre requisitos (eje Y) que se han realizado por semana (eje X).

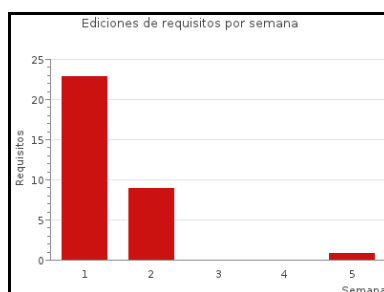


Ilustración 51. Ejemplo de gráfica de ediciones de requisitos por semana.

- Estabilidad media de requisitos por semana: Gráfica de barras que indica por cada semana (eje X) la estabilidad media conjunta de todos los requisitos (eje Y). La estabilidad media es calculada realizando la media aritmética de la estabilidad de cada requisito. La estabilidad de cada requisito se calcula por cada edición que sufre de la siguiente forma: $Estabilidad = 100 - (N_ediciones_req / N_ediciones_totales)$, donde $N_ediciones_req$ es el número de ediciones del requisito editado y $N_ediciones_totales$ es el número de ediciones de todos los requisitos. Así, un requisito recién creado será 100% estable, pero conforme se edita y se generan nuevas versiones del mismo, la estabilidad irá disminuyendo teniendo en cuenta las ediciones del resto de requisitos.

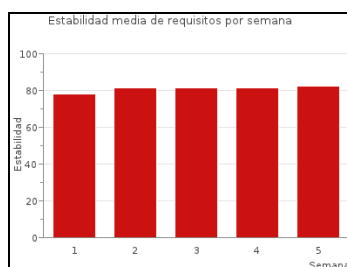


Ilustración 52. Ejemplo de gráfica de estabilidad de requisitos por semana.

- Número de iteraciones por estado: gráfica del tipo circular en 3D que muestra los porcentajes del número de iteraciones por los diferentes estados.



Ilustración 53. Ejemplo de gráfica de iteraciones por estado.

- Casos de uso por estado: gráfica del tipo circular en 3D que muestra los porcentajes del número de casos de uso por los diferentes estados.



Ilustración 54. Ejemplo de gráfica de casos de uso por estado.

- Entregables por estado: gráfica del tipo circular en 3D que muestra los porcentajes del número de entregables por los diferentes estados.



Ilustración 55. Ejemplo de gráfica de entregables por estado.

- Casos de uso editados por semana: Gráfica de barras que indica el número de ediciones sobre casos de uso (eje Y) que se han realizado por semana (eje X).

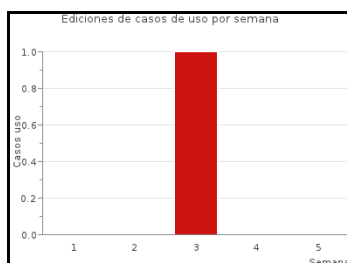


Ilustración 56. Ejemplo de gráfica de ediciones de casos de uso por semana.

- Entregables editados por semana: Gráfica de barras que indica el número de actualizaciones sobre entregables (eje Y) que se han realizado por semana (eje X).

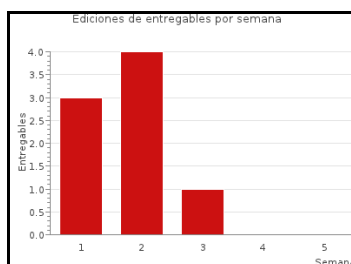


Ilustración 57. Ejemplo de gráfica de ediciones de entregables por semana.

- Requisitos por versión: gráfica del tipo circular en 3D que muestra los porcentajes del número de requisitos por los diferentes límites de versiones (1, de 2 a 5, de 6 a 10, de 11 a 50 y más de 50).

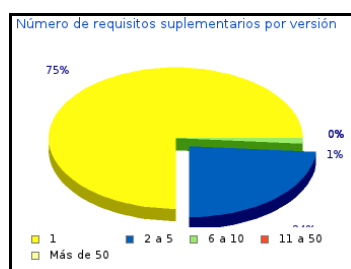


Ilustración 58. Ejemplo de gráfica de requisitos por versión.

- Casos de uso por versión: gráfica del tipo circular en 3D que muestra los porcentajes del número de casos de uso por los diferentes límites de versiones (1, de 2 a 5, de 6 a 10, de 11 a 50 y más de 50).

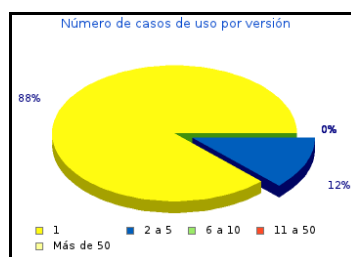


Ilustración 59. Ejemplo de gráfica de casos de uso por versión.

- Entregables por versión: gráfica del tipo circular en 3D que muestra los porcentajes del número de entregables por los diferentes límites de versiones (1, de 2 a 5, de 6 a 10, de 11 a 50 y más de 50).



Ilustración 60. Ejemplo de gráfica de entregables por versión.

Permisos

Permisos del sistema

El sistema CPS forma parte de un proyecto global, el cual se divide en dos ramas: gestión de usuarios y gestión de proyectos. Aunque CPS abarque la segunda rama, dispone de un pequeño módulo para interpretar la gestión de usuarios y roles de la primera rama.

Los usuarios que pertenezcan a la primera rama, tendrán acceso a CPS dependiendo de su perfil. En dicho perfil encontrarán su rol general, los proyectos en los que participa y el rol específico para cada proyecto. CPS hace uso de dicha información para clasificar así diferentes opciones y áreas dependiendo del usuario que accede al sistema.

Los roles que actualmente conoce el sistema CPS y por tanto hace uso de ellos para gestionar diferentes acciones y secciones son los siguientes:

- Administrador.
 - Administración total del sistema.
 - El usuario administrador puede acceder a todas las áreas del programa, así como realizar cualquier acción de lectura, creación, edición o borrado de cualquier elemento.
- Propietario.
 - Propietario del sistema en desarrollo.
 - Permiso de acceso únicamente a documentación publicada y al historial del proyecto.
 - Acciones permitidas: aceptación o rechazo de requisitos, casos de uso y entregables.
- Project Manager.
 - Jefe del proyecto en desarrollo.
 - El usuario puede acceder a todas las áreas dentro del proyecto al que pertenezca.
 - Acciones permitidas:
 - Asignación de usuarios al proyecto.
 - Asignación de roles a los usuarios participantes.
 - Creación y edición del Plan de Iteraciones.
 - Lectura, creación, edición y publicación de requisitos, casos de uso y entregables.

- Descarga de entregables y del proyecto completo.
- Edición de las propiedades del proyecto.
- Ingeniero de Requisitos.
 - Encargado de la gestión de los requisitos y casos de uso del proyecto.
 - Acceso a todas las áreas del proyecto en modo lectura, excepto "Usuarios" y "Propiedades".
 - Acciones permitidas:
 - Lectura del Plan de Iteraciones.
 - Lectura, creación, edición y publicación al Project Manager de requisitos y casos de uso.
 - Lectura de entregables.
- Analista.
 - Encargado de modelos de análisis. Funcionalidad no incluida en CPS.
 - Acceso al área "Requisitos", "Trazabilidad" y "Entregables" en modo lectura.
 - Acciones permitidas: lectura de requisitos, casos de uso y entregables.
- Gestor de Entregables.
 - Encargado de la gestión de los entregables del proyecto.
 - Acceso a todas las áreas del proyecto en modo lectura, excepto "Usuarios" y "Propiedades".
 - Acciones permitidas:
 - Lectura del Plan de Iteraciones.
 - Lectura, creación, edición y publicación al Project Manager de entregables.
 - Lectura de requisitos y casos de uso.

Avisos

Acceso a avisos

CPS dispone de un sistema de avisos desde el que los usuarios de un mismo proyecto pueden recibir notificaciones o comunicarse entre ellos.

El usuario, sea cual sea su rol, puede acceder a la sección de avisos desde cualquier ventana. Sólo debe seleccionar el desplegable de la derecha, situado en la barra superior, y hacer clic en la opción "Avisos". En la pantalla principal de avisos el usuario tendrá disponible el listado de avisos disponibles. Desde la misma pantalla podrá crear nuevos avisos, así como responder o eliminar avisos ya existentes.

Recepción de avisos

El programa CPS dispone de un envío automático de avisos dependiendo de la acción realizada. Las condiciones y avisos automáticos son los siguientes:



- Requisito, caso de uso o entregable enviado a evaluación: esta notificación es enviada en el momento en que un Ingeniero de Requisitos publica un requisito o caso de uso o un Gestor de Entregables publica un entregable. La notificación será enviada al Project Manager indicando qué elemento ha sido enviado para su evaluación, así como el responsable de su publicación y el nombre del proyecto al que pertenezca el elemento.
- Requisito, caso de uso o entregable publicado al cliente: la notificación es enviada en el momento en que un Project Manager publica alguno de estos elementos. Dicha notificación será recibida por los clientes del proyecto, indicando qué elemento ha sido publicado, de qué proyecto y quién lo ha publicado.
- Requisito, caso de uso o entregable rechazado por un cliente: la notificación es enviada a los Project Manager del proyecto del elemento rechazado. Dicho aviso mostrará el nombre del cliente que lo ha rechazado, el proyecto en el que se encuentra, el elemento rechazado y las razones del rechazo.
- Requisito, caso de uso o entregable aceptado por un cliente: la notificación es enviada a los Project Manager del proyecto del elemento aceptado. Dicho aviso mostrará el nombre del cliente que lo ha aceptado, el proyecto en el que se encuentra y el elemento aceptado.

Remitente	Proyecto	Mensaje	Fecha	
Sr. Administrador Apellido	test	Test	2012-03-06 09:57:35	
Luis Miguel Sánchez García	CPS	Requisito aceptado: Gestionar proyectos	2012-03-20 14:54:48	
Luis Miguel Sánchez García	CPS	Requisito aceptado: Gestionar iteraciones	2012-03-20 14:54:52	
Luis Miguel Sánchez García	CPS	Requisito aceptado: Gestionar requisitos	2012-03-20 14:54:57	
Luis Miguel Sánchez García	CPS	Requisito aceptado: Gestionar casos de uso	2012-03-20 14:55:05	

Ilustración 61. Avisos recibidos.

Aparte de los avisos automáticos, el usuario podrá recibir otros avisos creados por otros usuarios. En todos los avisos, se muestra la fecha y hora del envío.



Por cada aviso, el usuario podrá tomar las siguientes opciones: [leer](#), [responder](#) o [eliminar](#).

Leer aviso

El usuario podrá leer el contenido del mensaje directamente desde el listado de los avisos, pero si lo prefiere podrá hacer clic sobre el aviso para acceder a la lectura individual del aviso seleccionado. Dicha pantalla mostrará el emisor del aviso y el mensaje.

El usuario dispone también del botón "Responder".

Respuesta a aviso

Si el usuario selecciona la opción "Responder" desde el [listado de avisos](#) o desde la pantalla de lectura de un aviso, accederá a la pantalla de creación de aviso. A diferencia de la opción "Crear nuevo aviso", en este caso el destinatario no se puede modificar, ya que es el emisor del aviso a responder.

Lo único que deberá hacer el usuario es escribir su mensaje y hacer clic en el botón "Aceptar".

Envío de avisos

Desde [la pantalla principal de la sección de avisos](#), el usuario tiene disponible el enlace "Nuevo aviso" para escribir un nuevo aviso.

La comunicación entre usuarios tiene una restricción: deben pertenecer a un mismo proyecto. Por ello, si el usuario no ha seleccionado ningún proyecto, el sistema no le permitirá crear ningún aviso. En caso contrario, el usuario accederá a la pantalla de creación de nuevo aviso. En dicha pantalla, el usuario deberá rellenar dos campos:

- Para: es el destinatario del mensaje. El listado muestra los usuarios que pertenecen al mismo proyecto.
- Mensaje: el cuerpo del mensaje a enviar.

Ambos campos son obligatorios. Una vez cumplimentados, el usuario podrá hacer clic en el botón "Aceptar" para enviar el mensaje.

Eliminar avisos

Desde [la pantalla principal de avisos](#), el usuario podrá hacer clic en la opción "Eliminar" (icono de cruz roja) para eliminar cualquier aviso. Por seguridad, el sistema le pedirá una segunda confirmación.

Anexo IV: Funciones auxiliares.

FUNCIÓN	DESCRIPCIÓN	PARÁMETROS DE ENTRADA	PARÁMETROS DE SALIDA
conectar()	Crea una conexión con el servidor MySQL. Para ello, utiliza la función <code>mysql_connect(...)</code> con los datos proporcionados por el usuario a la hora de instalar el sistema.	Ninguno.	La instancia de la conexión si se ha realizado correctamente. False en caso contrario.
seleccionarBD()	Selecciona la base de datos del sistema utilizando la función <code>mysql_select_db(...)</code> .	Ninguno.	True si se ha seleccionado la base de datos correctamente. False en caso contrario.
consultar(consulta, conexionBD)	Realiza una consulta a la base de datos con la que se ha establecido previamente una conexión.	- consulta: cadena de texto con la consulta sql a ejecutar. - conexionBD: la conexión establecida con la base de datos.	El resultado de la consulta o false en caso de error en la misma.
enviarConsultaUnica(sentencia)	Envía una consulta a la base de datos cuya petición hace referencia a un único resultado (consulta de tipo SELECT).	- sentencia: la consulta a enviar a la base de datos.	El resultado de la consulta o false en caso contrario.
enviarInsert(sentencia)	Envía una sentencia de inserción a la base de datos (consulta de tipo INSERT).	- sentencia: la consulta a enviar a la base de datos.	True si la sentencia se ha ejecutado correctamente o false en caso contrario.
enviarConsultaMultiple(sentencia)	Envía una consulta a la base de datos cuya petición hace referencia a un conjunto de resultados (consulta de tipo SELECT).	- sentencia: la consulta a enviar a la base de datos.	Array con el resultado de la consulta. False si ha ocurrido algún error.
getTamanyo(array)	Calcula el número de elementos de un array. Para ello, utiliza la función <code>count(...)</code> .	- array: el array a comprobar.	Número entero con el tamaño del array.
desconectar(conexion)	Desconecta la aplicación de la base de datos MySQL. Utiliza para ello la función <code>mysql_close(...)</code> .	- conexión: la conexión establecida anteriormente con la base de datos.	True si se ha realizado la acción correctamente. False en caso contrario.
fechaActual()	Devuelve la hora actual del sistema en formato HH:MM.	Ninguno.	La fecha actual en el formato indicado.
diaActual()	Devuelve la fecha actual del sistema en formato aaaa-mm-dd	Ninguno.	La fecha actual en el formato

			indicado.
generaPass(long)	Genera un texto aleatorio de longitud indicada.	- long: la longitud del texto a generar.	La cadena de texto generada.
enviarMail(destino, asunto, cuerpo, cabeceras)	Envía un correo electrónico con los datos indicados al destinatario indicado.	- destino: el correo electrónico del destinatario. - asunto: el asunto del correo electrónico. - cuerpo: el mensaje del correo electrónico. - cabeceras: las cabeceras del correo electrónico.	True si se ha enviado correctamente. False en caso contrario.
identificarUsuario(user, pass)	Verifica que los credenciales indicados son los correctos en la base de datos.	- user: el nombre de usuario. - pass: la contraseña.	True si ha sido identificado correctamente. False en caso contrario.
existeUsuario(user)	Comprueba si el nombre de usuario ya existe en la base de datos.	- user: el nombre de usuario a comprobar.	True si el usuario existe. False en caso contrario.
registrar(nombre, apellidos, user, pass)	Crea un nuevo usuario en la base de datos.	- nombre: el nombre propio del usuario a registrar. - apellidos: los apellidos del usuario. - user: el nombre de usuario. - pass: la contraseña.	True si se ha realizado la acción correctamente. False en caso contrario.
modificar(nombre, apellidos, user, pass)	Modifica un usuario existente en la base de datos.	- nombre: el nombre propio del usuario a registrar. - apellidos: los apellidos del usuario. - user: el nombre de usuario. - pass: la contraseña.	True si se ha realizado la acción correctamente. False en caso contrario.
crear_proyecto(nuevo_nombre_proyecto, nueva_descripcion_proyecto)	Crea un nuevo proyecto en la base de datos.	- nuevo_nombre_proyecto: el título del nuevo proyecto. - nueva_descripcion_proyecto: la descripción del nuevo proyecto.	True si se ha realizado la acción correctamente. False en caso contrario.
getInicioProyecto(id_proyecto)	Obtiene la fecha de creación de un proyecto.	- id_proyecto: el identificador del proyecto a comprobar.	La fecha en la que se creó el proyecto.
pasar_fase(id_proyecto, id_fase, id_usuario_accion)	Aumenta en una unidad la fase actual de un proyecto.	- id_proyecto: el identificador del proyecto. - id_fase: la fase nueva del proyecto. - id_usuario_accion: el identificador del usuario que realiza la acción.	True si se ha realizado la acción correctamente. False en caso contrario.
pasar_iteracion(id_proyecto, id_it, id_usuario_accion)	Aumenta en una unidad la iteración actual de un proyecto.	- id_proyecto: el identificador del proyecto. - id_it: la iteración nueva del proyecto. - id_usuario_accion: el identificador del usuario que realiza la acción.	True si se ha realizado la acción correctamente. False en caso contrario.
editar_proyecto(id_proyecto, nuevo_nombre_pro)	Edita los datos de un proyecto existente.	- id_proyecto: el identificador del proyecto a editar. - nuevo_nombre_proyecto: el nuevo nombre del proyecto (o	True si se ha realizado la acción correctamente.

ycto, nueva_descripcion_ proyecto, id_usuario_accion)		el anterior si no se desea modificar). - nueva_descripcion_proyecto: la nueva descripción del proyecto (o la anterior si no se desea modificar). - id_usuario_accion: el identificador del usuario que realiza la acción.	False en caso contrario.
crear_requisito(titulo, descripcion, tipo, subtipo, impacto, esfuerzo, id_proyecto, id_usuario, trazabilidad)	Crea un nuevo requisito en la base de datos con los datos indicados.	- titulo: el título del nuevo requisito. - descripcion: la descripción del nuevo requisito. - tipo: el tipo de requisito a crear. - subtipo: el subtipo del tipo indicado (si es opcional, indicar cadena de caracteres vacía). - impacto: identificador del nivel de impacto. - esfuerzo: identificador del nivel de esfuerzo. - id_proyecto: identificador del proyecto al que pertenecerá el nuevo requisito. - id_usuario: identificador del usuario que crea el requisito. - trazabilidad: tipo de trazabilidad del requisito.	True si se ha realizado la acción correctamente. False en caso contrario.
crear_entregable(titulo, descripcion, archivos_adjuntos, num_archivos, id_proyecto, id_usuario, id_producto)	Crea un nuevo entregable en la base de datos con los datos indicados. Esta función también genera el archivo asociado.	- titulo: título del nuevo entregable. - descripcion: descripción del entregable. - archivos_adjuntos: array con los archivos que se añadirán al entregable. - num_archivos: longitud del array anterior. - id_proyecto: identificador del proyecto al que pertenecerá el entregable. - id_usuario: identificador del usuario que genera el entregable. - id_producto: identificador del producto que contiene el nombre del entregable.	True si se ha realizado la acción correctamente. False en caso contrario.
actualizar_entregable(id_entregable, archivos_adjuntos, num_archivos, id_proyecto, id_usuario)	Actualiza un entregable existente en la base de datos.	- id_entregable: identificador del entregable a actualizar. - archivos_adjuntos: array de documentos que se añadirán al entregable indicado. - num_archivos: longitud del array anterior. - id_proyecto: identificador del proyecto al que pertenece el entregable. - id_usuario: identificador del usuario que actualiza el entregable.	True si se ha realizado la acción correctamente. False en caso contrario.
existeTipo(tipo)	Comprueba si el tipo de requisito indicado ya existe en la base de datos.	- tipo: nombre del tipo de requisito a comprobar.	True si el tipo existe. False en caso contrario.

existeSubtipo(subtipo, id_tipo)	Comprueba si el subtipo perteneciente al tipo indicado, ya existe en la base de datos.	<ul style="list-style-type: none"> - subtipo: nombre del subtipo a comprobar. - id_tipo: identificador del tipo padre del subtipo. 	True si el subtipo existe. False en caso contrario.
calcularEstabilidad(id_proyecto, id_requisito)	Calcula la estabilidad de un requisito perteneciente a un proyecto.	<ul style="list-style-type: none"> - id_proyecto: identificador del proyecto al que pertenece el requisito que se indica a continuación. - id_requisito: identificador del requisito a comprobar su estabilidad. 	Estabilidad del requisito expresada en tanto por ciento.
editar_requisito(id_requisito, id_tipo, titulo, descripcion, tipo, subtipo, impacto, esfuerzo, id_proyecto, id_usuario, trazabilidad)	Edita los datos de un requisito existente en la base de datos.	<ul style="list-style-type: none"> - id_requisito: identificador del requisito a editar. - titulo: el nuevo título del requisito. - descripción: la nueva descripción del requisito. - tipo: el nuevo tipo del requisito. - subtipo: el nuevo subtipo de requisito. - impacto: el nuevo impacto del requisito. - esfuerzo: el nuevo esfuerzo del requisito. - id_proyecto: identificador del proyecto al que pertenece el requisito. - id_usuario: identificador del usuario que edita el requisito. 	True si se ha realizado la acción correctamente. False en caso contrario.
crear_caso(titulo, ruta_completa, id_proyecto, id_usuario)	Crea un nuevo caso de uso en la base de datos.	<ul style="list-style-type: none"> - titulo: título del nuevo caso de uso. - ruta_completa: ruta donde se encuentra la imagen que representa el diagrama del caso de uso. - id_proyecto: identificador del proyecto al que pertenecerá el nuevo caso de uso. - id_usuario: identificador del usuario que crea el nuevo caso de uso. 	True si se ha realizado la acción correctamente. False en caso contrario.
editar_caso(id_caso, titulo, ruta_completa, id_proyecto, id_usuario)	Edita los datos de un caso de uso existente en la base de datos.	<ul style="list-style-type: none"> - id_caso: identificador del caso de uso a editar. - titulo: el nuevo título del caso de uso. - ruta_completa: la nueva ruta de la imagen que representa el diagrama del caso de uso. - id_proyecto: el identificador del proyecto al que pertenece el caso de uso. - id_usuario: identificador del usuario que edita el proyecto. 	True si se ha realizado la acción correctamente. False en caso contrario.
crear_iteracion(id_proyecto, id_fase, nombre, duracion, id_usuario_accion)	Crea una nueva iteración en la base de datos.	<ul style="list-style-type: none"> - id_proyecto: identificador del proyecto al que pertenecerá la nueva iteración. - id_fase: identificador de la fase a la que pertenecerá la nueva iteración. - nombre: el nombre de la nueva iteración. - duración: la duración en semanas de la nueva iteración (número entero). - id_usuario_accion: el 	True si se ha realizado la acción correctamente. False en caso contrario.

		identificador del usuario que crea la iteración.	
editar_iteracion(id_proyecto, id_fase, id_it, nombre, duracion, id_usuario_accion)	Edita los datos de una iteración existente en la base de datos.	<ul style="list-style-type: none"> - id_proyecto: el identificador del proyecto al que pertenece la iteración a editar. - id_fase: el identificador de la fase a la que pertenece la iteración a editar. - id_it: el identificador de la iteración a editar. - nombre: el nuevo nombre de la iteración. - duracion: la nueva duración en semanas de la iteración (número entero). - id_usuario_accion: el identificador del usuario que edita la iteración. 	True si se ha realizado la acción correctamente. False en caso contrario.
existeProductoIteracion(id_proyecto, id_fase, id_it, id_producto, id_responsable)	Comprueba si un producto está asociado a una determinada iteración dentro de un proyecto.	<ul style="list-style-type: none"> - id_proyecto: identificador del proyecto al que pertenece la iteración a comprobar. - id_fase: identificador de la fase en la que pertenece la iteración a comprobar. - id_it: identificador de la iteración de la que se desea comprobar un producto. - id_producto: identificador del producto a comprobar en la iteración. - id_responsable: identificador del usuario responsable del producto a comprobar. 	True si el producto existe en la iteración. False en caso contrario.
asignar_iteracion_producto(id_proyecto, id_fase, id_it, id_producto, id_usuario_accion, id_responsable)	Crea una relación entre un producto y una iteración de un proyecto.	<ul style="list-style-type: none"> - id_proyecto: identificador del proyecto al que pertenece la iteración donde se asignará el producto. - id_fase: identificador de la fase a la que pertenece la iteración donde se asignará el producto. - id_it: identificador de la iteración donde se asignará el producto. - id_producto: identificador del producto a asignar en la iteración. - id_usuario_accion: identificador del usuario que realiza la asignación. - id_responsable: identificador del usuario responsable del producto a asignar. 	True si se ha realizado la acción correctamente. False en caso contrario.
desasignar_iteracion_producto(id_proyecto, id_fase, id_it, id_producto, id_usuario_accion)	Elimina la relación entre un producto y una iteración de un proyecto.	<ul style="list-style-type: none"> - id_proyecto: identificador del proyecto al que pertenece la iteración donde se desasignará el producto. - id_fase: identificador de la fase a la que pertenece la iteración donde se desasignará el producto. - id_it: identificador de la iteración donde se desasignará el producto. - id_producto: identificador del producto a desasignar de la iteración. - id_usuario_accion: identificador del usuario que realiza la desasignación. 	True si se ha realizado la acción correctamente. False en caso contrario.

		identificador del usuario que desasigna el producto.	
array_envia(array)	Codifica un array de datos en una cadena de texto para pasarla por parámetro de forma segura.	- array: el array a codificar.	Array codificado.
array_recibe(url_array)	Decodifica un array de datos codificado.	- url_array: el array codificado.	Array decodificado.
getPos(id_campo, id_proyecto, campo, tabla)	Obtiene el identificador único del campo indicado en un proyecto específico.	- id_campo: identificador del elemento del que se quiere obtener su posición. - id_proyecto: identificador del proyecto al que pertenece el elemento anterior. - campo: nombre del campo por el que se desea ordenar a los elementos de la consulta. - tabla: nombre de la tabla de la base de datos donde se encuentra el elemento.	Posición del elemento.
existeUsuarioProyecto(id_proyecto, id_usuario)	Comprueba si un usuario pertenece a un proyecto.	- id_proyecto: identificador del proyecto en el que se desea comprobar la existencia de un usuario. - id_usuario: identificador del usuario a comprobar.	True si el usuario existe. False en caso contrario.
tieneTrazabilidad(id_requisito, id_caso)	Comprueba si un requisito dispone de trazabilidad con un caso de uso.	- id_requisito: el identificador del requisito. - id_caso: el identificador del caso de uso.	True si el requisito dispone de trazabilidad con el caso de uso. False en caso contrario.
getIdReq(texto)	Extrae el identificador de un requisito desde una cadena de texto.	- texto: la cadena de texto desde la que se extraerá el identificador del requisito.	Identificador del requisito. Cadena vacía si no se ha encontrado.
getIdCaso(texto)	Extrae el identificador de un caso de uso desde una cadena de texto.	- texto: la cadena de texto desde la que se extraerá el identificador del caso de uso.	Identificador del caso de uso. Cadena vacía si no se ha encontrado.
trazar(id_req, id_caso)	Crea una dependencia de trazabilidad entre un requisito y un caso de uso.	- id_req: identificador del requisito. - id_caso: identificador del caso de uso.	True si se ha realizado la acción correctamente. False en caso contrario.
eliminar_traza(id_req, id_caso)	Elimina una relación de trazabilidad entre un requisito y un caso de uso.	- id_req: identificador del requisito. - id_caso: identificador del caso de uso.	True si se ha realizado la acción correctamente. False en caso contrario.

asignar_proyecto(id_usuario_accion, id_proyecto, id_usuario, accion, fecha, id_rol)	Asigna, desasigna o modifica a un usuario de un proyecto con un determinado rol.	<ul style="list-style-type: none"> - id_usuario_accion: identificador del usuario que realiza la acción. - id_proyecto: identificador del proyecto al que se asignará/desasignará el nuevo usuario. - id_usuario: identificador del usuario que se asignará/desasignará al/del proyecto. - accion: "Alta", "Modificación", "Baja". - fecha: fecha en la que se realiza la acción. - id_rol: identificador del rol del usuario participante. 	True si se ha realizado la acción correctamente. False en caso contrario.
crearAviso(id_proyecto, id_emisor, id_receptor, mensaje)	Crea un nuevo aviso en la base de datos.	<ul style="list-style-type: none"> - id_proyecto: identificador del proyecto al que pertenecerá el nuevo aviso. - id_emisor: identificador del usuario que crea el aviso. - id_receptor: identificador del usuario que recibirá el aviso. - mensaje: el cuerpo del aviso. 	True si se ha realizado la acción correctamente. False en caso contrario.
eliminarAviso(id_aviso)	Elimina un aviso existente en la base de datos.	<ul style="list-style-type: none"> - id_aviso: identificador del aviso a eliminar. 	True si se ha realizado la acción correctamente. False en caso contrario.
marcarLeido(id_aviso)	Marca como leído un aviso.	<ul style="list-style-type: none"> - id_aviso: identificador del aviso que se marcará como leído. 	True si se ha realizado la acción correctamente. False en caso contrario.
publicarCasoCliente(id_caso, titulo, id_usuario, id_proyecto)	Publica un caso de uso al propietario del proyecto.	<ul style="list-style-type: none"> - id_caso: identificador del caso de uso a publicar. - titulo: nombre del caso de uso a publicar. - id_usuario: identificador del usuario que publica el caso de uso. - id_proyecto: identificador del proyecto al que pertenece el caso de uso a publicar. 	True si se ha realizado la acción correctamente. False en caso contrario.
publicarCasoPM(id_caso, titulo, id_usuario, id_proyecto)	Manda a revisión un caso de uso al jefe del proyecto.	<ul style="list-style-type: none"> - id_caso: identificador del caso de uso a publicar. - titulo: nombre del caso de uso a publicar. - id_usuario: identificador del usuario que publica el caso de uso - id_proyecto: identificador del proyecto al que pertenece el caso de uso a publicar. 	True si se ha realizado la acción correctamente. False en caso contrario.
publicarRequisitoCliente(id_requisito, titulo, id_usuario, id_proyecto)	Publica un requisito al propietario del proyecto.	<ul style="list-style-type: none"> - id_requisito: identificador del requisito a publicar. - titulo: nombre del requisito a publicar. - id_usuario: identificador del usuario que publica el requisito. - id_proyecto: identificador del proyecto al que pertenece el 	True si se ha realizado la acción correctamente. False en caso contrario.

		requisito a publicar.	
publicarRequisitoPM(id_requisito, titulo, id_usuario, id_proyecto)	Manda a revisión un requisito al jefe del proyecto.	- id_requisito: identificador del requisito a publicar. - titulo: nombre del requisito a publicar. - id_usuario: identificador del usuario que publica el requisito. - id_proyecto: identificador del proyecto al que pertenece el requisito a publicar.	True si se ha realizado la acción correctamente. False en caso contrario.
aceptarCaso(id_caso, titulo, id_usuario, id_proyecto)	Acepta un caso de uso.	- id_caso: identificador del caso de uso a aceptar. - titulo: nombre del caso de uso a aceptar. - id_usuario: identificador del usuario que realiza la acción. - id_proyecto: identificador del proyecto al que pertenece el caso de uso.	True si se ha realizado la acción correctamente. False en caso contrario.
rechazarCaso(id_caso, titulo, id_usuario, id_proyecto)	Rechaza un caso de uso.	- id_caso: identificador del caso de uso a rechazar. - titulo: nombre del caso de uso a rechazar. - id_usuario: identificador del usuario que realiza la acción. - id_proyecto: identificador del proyecto al que pertenece el caso de uso.	True si se ha realizado la acción correctamente. False en caso contrario.
aceptarRequisito(id_requisito, titulo, id_usuario, id_proyecto)	Acepta un requisito.	- id_requisito: identificador del requisito a aceptar. - titulo: nombre del requisito a aceptar. - id_usuario: identificador del usuario que realiza la acción. - id_proyecto: identificador del proyecto al que pertenece el requisito.	True si se ha realizado la acción correctamente. False en caso contrario.
rechazarRequisito(id_requisito, titulo, id_usuario, id_proyecto)	Rechaza un requisito.	- id_requisito: identificador del requisito a rechazar. - titulo: nombre del requisito a rechazar. - id_usuario: identificador del usuario que realiza la acción. - id_proyecto: identificador del proyecto al que pertenece el requisito.	True si se ha realizado la acción correctamente. False en caso contrario.
publicarEntregableCliente(id_entregable, titulo, id_usuario, id_proyecto)	Publica un entregable al propietario del proyecto.	- id_entregable: identificador del entregable a publicar. - titulo: nombre del entregable a publicar. - id_usuario: identificador del usuario que publica el entregable. - id_proyecto: identificador del proyecto al que pertenece el entregable a publicar.	True si se ha realizado la acción correctamente. False en caso contrario.



publicarEntregablePM(id_entregable, titulo, id_usuario, id_proyecto)	Manda a revisión un entregable al jefe del proyecto.	- id_entregable: identificador del entregable a publicar. - titulo: nombre del entregable a publicar. - id_usuario: identificador del usuario que publica el entregable. - id_proyecto: identificador del proyecto al que pertenece el entregable a publicar.	True si se ha realizado la acción correctamente. False en caso contrario.
aceptarEntregable(id_entregable, titulo, id_usuario, id_proyecto)	Acepta un entregable.	- id_entregable: identificador del entregable a aceptar. - titulo: nombre del entregable a aceptar. - id_usuario: identificador del usuario que realiza la acción. - id_proyecto: identificador del proyecto al que pertenece el entregable.	True si se ha realizado la acción correctamente. False en caso contrario.
rechazarEntregable(id_entregable, titulo, id_usuario, id_proyecto)	Rechaza un entregable.	- id_entregable: identificador del entregable a rechazar. - titulo: nombre del entregable a rechazar. - id_usuario: identificador del usuario que realiza la acción. - id_proyecto: identificador del proyecto al que pertenece el entregable.	True si se ha realizado la acción correctamente. False en caso contrario.
eliminarProyecto(id)	Elimina un proyecto de la base de datos.	- id: identificador del proyecto a eliminar.	True si se ha realizado la acción correctamente. False en caso contrario.
eliminarRequisito(id_proyecto, id, id_usuario_accion)	Elimina un requisito existente en la base de datos.	- id_proyecto: identificador del proyecto al que pertenece el requisito a eliminar. - id: identificador del requisito a eliminar. - id_usuario_accion: identificador del usuario que elimina el requisito.	True si se ha realizado la acción correctamente. False en caso contrario.
eliminarCasoUso(id_proyecto, id, id_usuario_accion)	Elimina un caso de uso existente en la base de datos.	- id_proyecto: identificador del proyecto al que pertenece el caso de uso a eliminar. - id: identificador del caso de uso a eliminar. - id_usuario_accion: identificador del usuario que elimina el caso de uso.	True si se ha realizado la acción correctamente. False en caso contrario.
eliminarIteracion(id_proyecto, id, id_fase, id_usuario_accion)	Elimina una iteración existente en la base de datos.	- id_proyecto: identificador del proyecto al que pertenece la iteración a eliminar. - id: identificador de la iteración a eliminar. - id_fase: identificador de la fase en la que pertenece la iteración a eliminar. - id_usuario_accion: identificador del usuario que elimina la iteración.	True si se ha realizado la acción correctamente. False en caso contrario.

eliminarEntregable(id_proyecto, id_usuario_accion)	Elimina un entregable existente en la base de datos.	- id_proyecto: identificador del proyecto al que pertenece el entregable a eliminar. - id: identificador del entregable a eliminar. - id_usuario_accion: identificador del usuario que elimina el entregable.	True si se ha realizado la acción correctamente. False en caso contrario.
activarGrafica(id_usuario, id_proyecto, id_grafica)	Activa una gráfica para un usuario y en un proyecto concreto.	- id_usuario: identificador del usuario que realiza la activación. - id_proyecto: identificador del proyecto sobre el que se realiza la activación. - id_grafica: identificador de la gráfica que se activa.	True si se ha realizado la acción correctamente. False en caso contrario.
desactivarGrafica(id_usuario, id_proyecto, id_grafica)	Desactiva una gráfica para un usuario y en un proyecto concreto.	- id_usuario: identificador del usuario que realiza la desactivación. - id_proyecto: identificador del proyecto sobre el que se realiza la desactivación. - id_grafica: identificador de la gráfica que se desactiva.	True si se ha realizado la acción correctamente. False en caso contrario.
graficaActivada(id_proyecto, id_usuario, id_grafica)	Comprueba si un usuario, en un proyecto en concreto, tiene activada una gráfica.	- id_proyecto: identificador del proyecto sobre el que se realiza la comprobación. - id_usuario: identificador del usuario que realiza la comprobación. - id_grafica: identificador de la gráfica que se comprueba.	True si se ha realizado la acción correctamente. False en caso contrario.
algunaGraficaActivada(id_usuario, id_proyecto)	Comprueba si un usuario tiene alguna gráfica activada en un proyecto concreto.	- id_usuario: identificador del usuario a comprobar. - id_proyecto: identificador del proyecto a comprobar.	True si se ha realizado la acción correctamente. False en caso contrario.
verificarBarra(ruta)	Verifica que la ruta indicada contiene una barra como último carácter. De no ser así, se añade la barra.	- ruta: ruta completa a verificar.	La ruta con la barra añadida.
crearArchivoConfiguracion(nombre_bbdd, nombre_mysql, nombre_ruta_server)	Crea el archivo de configuración con los datos indicados.	- nombre_bbdd: Nombre de la base de datos. - nombre_mysql: Ruta del servidor MySQL. - nombre_ruta_server: Ruta donde se encuentra instalada la aplicación.	True si se ha realizado la acción correctamente. False en caso contrario.
parsear(text)	Codifica el texto indicado eliminando tildes y caracteres especiales.	- text: cadena de texto a modificar.	El texto modificado.
crearDOC(título, autor, adjuntos, num_adjuntos, id_entregable, id_proyecto, id_producto)	Crea un documento de tipo .doc con los datos indicados.	- título: el título del documento. - autor: el autor del documento. - adjuntos: array de documentos que se añadirán al documento a crear. - num_adjuntos: tamaño del array anterior. - id_entregable: identificador	La ruta del nuevo documento.

		del entregable cuyo documento se va a crear. - id_proyecto: identificador del proyecto al que pertenecerá el nuevo documento. - id_producto: identificador del tipo de producto que será el nuevo documento.	
actualizarDOC(título, autor, adjuntos, num_adjuntos, ruta_entregable, id_proyecto)	Actualiza un documento de tipo .doc existente.	- título: el título del documento. - autor: el autor del documento. - adjuntos: array de documentos que se añadirán al documento a actualizar. - ruta_entregable: ruta del documento a actualizar. - id_proyecto: identificador del proyecto al que pertenecerá el nuevo documento.	La ruta del documento actualizado.
obtenerContenidoEntregable(id_producto)	Obtiene el contenido de un entregable a través de los datos existentes en la base de datos.	- id_producto: identificador del producto cuyo contenido se va a obtener.	El contenido del entregable.
leerAnteriorEntregable(ruta_entregable)	Obtiene el contenido de un entregable.	- ruta_entregable: ruta del documento a leer.	El contenido del entregable.
contenidoPlanDesarrollo()	Obtiene los datos del plan de iteraciones de un proyecto.	Ninguno.	El contenido obtenido de los datos del proyecto.
contenidoModeloCasosUso()	Obtiene los datos de los casos de uso de un proyecto.	Ninguno.	El contenido obtenido de los datos del proyecto.
contenidoRequisitosSuplementarios()	Obtiene los datos de los requisitos de un proyecto.	Ninguno.	El contenido obtenido de los datos del proyecto.
getNivel(id)	Obtiene el nombre del nivel indicado.	- id: identificador del nivel a obtener.	El nombre del nivel.
quitarAcentos(text)	Elimina las tildes del texto indicado.	- text: texto a modificar.	El texto modificado.
stri_replace(find, replace, string)	Del texto indicado, sustituye una cadena de texto por otra.	- find: cadena de texto que se reemplazará. - replace: cadena de texto que reemplazará a la anterior. - string: cadena de texto completa sobre la que se realizará la sustitución de subcadenas.	La cadena de texto modificada.
txt2html(txt)	Convierte texto en plano a formato HTML.	- txt: cadena de texto a convertir.	El texto convertido.
pdf2text(pdfdata)	Convierte el contenido de un fichero .PDF a HTML.	- pdfdata: contenido a convertir.	El texto convertido.



docx2text(filename)	Convierte el contenido de un fichero .DOCX a HTML.	- filename: Ruta del fichero a convertir.	El texto convertido.
readZippedXML(archiveFile, dataFile)	Lee el contenido de un fichero XML codificado.	- archiveFile: Ruta del fichero a leer. - dataFile: tipo de fichero (ej. "word/document.xml").	El texto leído.

Tabla 116. Funciones auxiliares.

Anexo V: Ejemplo práctico de uso.

Para comprobar el correcto funcionamiento de la aplicación, así como verificar su utilidad, se ha desarrollado el presente proyecto desde la propia aplicación.

Todas las ilustraciones incluidas en la sección “[Desarrollo](#)” han sido obtenidas de esta prueba real. Así se puede observar la utilidad de la aplicación frente a un proyecto real.

Proyectos:		
Título	Descripción	
CPS	Control Distribuido de Proyectos Software. Proyecto de Fin de Carrera de Adrián Hontoria.	

Ilustración 62. Desarrollo de un proyecto real.

A lo largo de la prueba de desarrollo, se han obtenido unos resultados muy buenos por parte del sistema. Gracias a él, se ha podido tener un control absoluto de todos los elementos del proyecto.

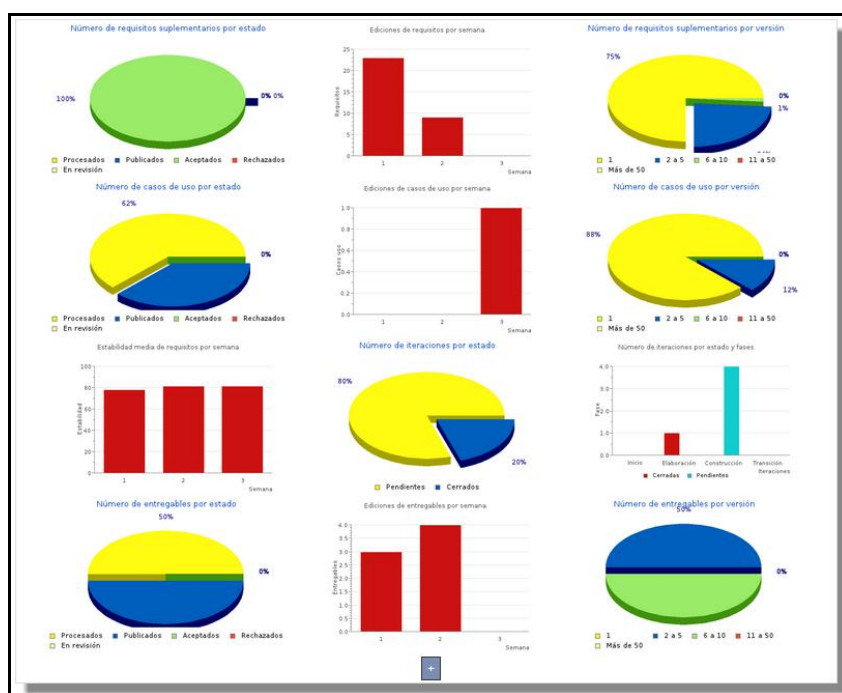


Ilustración 63. Control del desarrollo de un proyecto real.

De igual forma, gracias a la aplicación desarrollada se ha podido mantener un control real sobre el trabajo realizado en cada sección del proyecto.

Historial de CPS:

Realizado por: ▼▲	Acción realizada ▼▲	Acción aplicada a: ▼▲	Fecha ▼▲
Adrián Hontoria Sánchez	Editar caso de uso	Caso de uso Gestionar casos de uso	2012-04-09 06:29:25
Adrián Hontoria Sánchez	Publicar a cliente	Entregable Modelo de casos de uso	2012-04-09 06:27:49
Adrián Hontoria Sánchez	Publicar a cliente	Caso de uso Gestionar requisitos	2012-04-09 06:27:12
Adrián Hontoria Sánchez	Publicar a cliente	Caso de uso Gestionar iteraciones	2012-04-09 06:27:00
Adrián Hontoria Sánchez	Publicar a cliente	Caso de uso Gestionar proyectos	2012-04-09 06:26:52
Adrián Hontoria Sánchez	Actualizar	Entregable	2012-03-29 05:20:08
Adrián Hontoria Sánchez	Actualizar	Entregable	2012-03-29 05:20:02
Luis Miguel Sánchez García	Aceptar	Requisito Permiso de Gestor de entregables	2012-03-28 07:52:26
Luis Miguel Sánchez García	Aceptar	Requisito Gráfica "Entregables por estado"	2012-03-28 07:52:20
Adrián Hontoria Sánchez	Publicar a cliente	Requisito Permiso de Gestor de entregables	2012-03-28 07:51:20
Adrián Hontoria Sánchez	Publicar a cliente	Requisito Gráfica "Entregables por estado"	2012-03-28 07:50:55
Adrián Hontoria Sánchez	Editar requisito	Requisito Gráfica "Entregables por estado"	2012-03-28 07:50:04
Adrián Hontoria Sánchez	Editar requisito	Requisito Gráfica de "entregables por estado2"	2012-03-28 07:46:58
Adrián Hontoria Sánchez	Editar requisito	Requisito Permiso de Gestor de entregables	2012-03-28 07:44:32
Adrián Hontoria Sánchez	Crear	Caso de uso Gestionar avisos	2012-03-28 07:34:58

Anterior [Siguiente](#)

Ilustración 64. Control del trabajo realizado.

Finalmente, se puede concluir que la utilización del sistema en un proyecto real ha sido un éxito. Como prueba se pueden observar algunos productos generados automáticamente por el sistema.

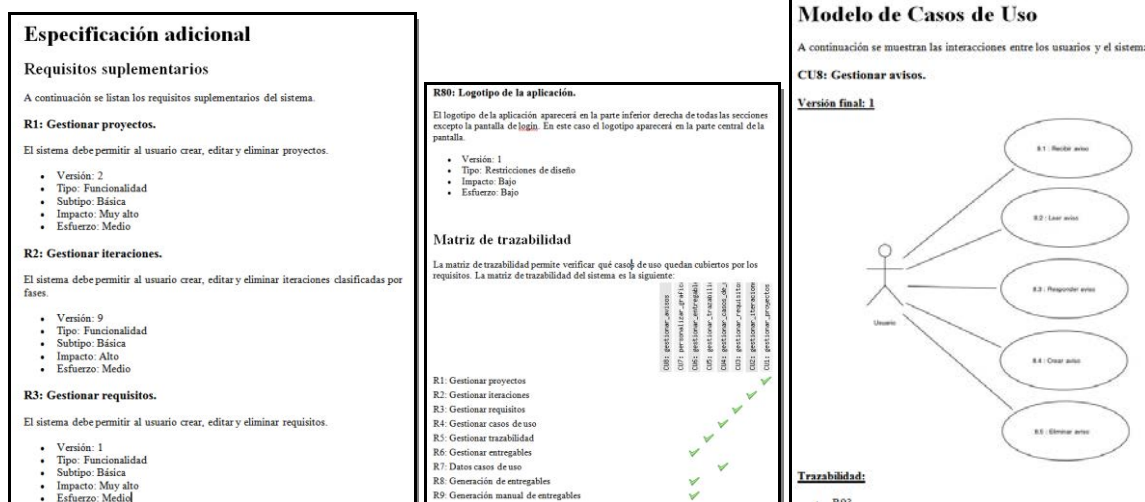


Ilustración 65. Productos generados del proyecto.